2000

Isolated word speech recognition using fuzzy neural techniques.

Hui. Ping
University of Windsor

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Isolated Word Speech Recognition Using Fuzzy Neural Techniques

by

Hui Ping

A Thesis
Submitted to the College of Graduate Studies and Research through the
Faculty of Engineering - Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

1999

Abstract

Automatic speech recognition by machine is one of the most efficient methods for man-machine communications. Because speech waveform is nonlinear and variant, speech recognition requires a lot of intelligence and fault tolerance in the pattern recognition algorithms. Fuzzy neural techniques allow effective decisions in the presence of uncertainty. Consequently, the objective of this thesis is to study the fuzzy neural techniques for the application in speech recognition. Two methods are proposed for isolated word recognition using fuzzy pattern matching technique and fuzzy c-means clustering technique. The algorithms are tested based on two LPC-based speech features: line spectrum frequencies and cepstral coefficients. It is shown that the fuzzy algorithm is an efficient approach and can provide reliable and accurate recognition results.

Dedicated to my family

for their love and support

Acknowledgements

I would like to express my sincere gratitude to my thesis advisor Dr. H. K. Kwan, for his suggestions, guidance, support and encouragement throughout the course of this research work. It has indeed been a privilege to work with him.

I wish to thank my department reader, Professor P. H. Alexander and my external reader, Dr. Liwu Li, for their valuable advice toward the fulfillment of the thesis work.

I would also like to thank all my friends in the ISPLab who have given me support during the study and research: Tracy Li, Halima El-Khatib, Wayne Chiang, Walter Jin and Jie Zhang.

Table of Contents

List of Abbreviations

ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
FCM	Fuzzy C-Means
DTW	Dynamic Time Warping
FL	Fuzzy Logic
FLVQ	Fuzzy Learning Vector Quantization
FNN	Fuzzy Neural Network
HMM	Hidden Markov Model
LPC	Linear Predictive Coding
LSF	Line Spectrum Frequency
LVQ	Learning Vector Quantization
SOM	Self-Organizing Map

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Automatic speech recognition by machine has been a part of science fiction for many years. The early attempts were made in the 1950s by various researchers. In 1952, Davis, Biddulph and Balashek [27] designed the first isolated digit recognizer for a single speaker at the Bell Laboratories. This system used a simple pattern matching method with templates for each of the digits. Matching was performed with two parameters: a frequency cut based on separating the spectrum of the spoken digit into two bands and a fundamental frequency estimated by zero-crossing counting.

In 1961, Suzuki and Nakata [28] in Tokyo built a hardware vowel recognizer based on a filter bank spectrum analyzer. In 1962, Sakai and Doshita of Tokyo University designed a hardware phoneme recognizer. A hardware speech segmentor was used along with a zero-crossing analysis for different segments of the input speech to provide the recognition result.

Most of the above systems were implemented as electronics devices. However, speech recognition could never attract so much attention until the flourish of digital computers.

The first computer-based speech recognition system was carried out in the early 60s. Denes and Matthews [29] introduced the concept of time normalization in speech pattern matching. In 1968, Russian researcher Vintsyuk [30] proposed the idea of dynamic programming methods of time alignment for speech patterns with different lengths. The essence of this idea, which is called DTW (dynamic time warping), is still widely used for the current commercial products.

The 1970s and 1980s were very active periods for speech recognition with a series of important milestones:

- Pattern recognition algorithms were applied for the template-based isolated word recognition methods.
- Continuous speech from large vocabularies was understood based on the use of high level knowledge to compensate for the errors in phonetic approaches.
- Speech analysis method based on Linear Predictive Coding (LPC) was used instead of conventional methods such as FFT and filter banks.
- Statistical modeling such as the HMMs (Hidden Markov Model) were developed for continuous speech recognition
- The neural networks (back propagation, learning vector quantization) with efficient learning algorithms were proposed for speech pattern matching

In recent years the speech recognition technology have begun to enter the real world in our life. More and more advanced algorithms were adopted in this area. Fuzzy neural

techniques have also been applied to speech recognition and this field is growing and developing very fast.

1.2 Applications of Speech Recognition Technology

Currently, speech recognition systems are being developed for commercial applications. One of the successful speech recognition systems is the Voice Recognition Call Processing (VRCP) system from AT&T. VPCP has a five-word vocabulary, and automates operator assisted calls. AT&T also have a system known as Voice Interactive Phone (VIP), with seven spoken commands replacing the touch tone codes. In this system, 94% of users were comfortable with talking to the machine, and 84% of users preferred the VIP system than the present system.

With computers becoming ever present in business, education, and government, there is a tremendous market for faster, more efficient man-machine interfaces. In the future, we will be intensely using voice as input along with the keyboard and mouse. Most of the windows or other GUI operating systems-based applications will use speech recognition to accept voice commands and convert voice into text.

A summary of speech technology application areas are listed below:

- Computer engineering: building a natural language interface to the computer operating system or application software.

- Program Developers: use pre-recorded voice-macros while developing a computer program.
- Telephone commerce (to replace touch-tone): telephone banking using voice commands; order placement using voice to record incoming order data for the customer service representatives.
- Telephony: hands-free dialing; connecting caller through a company switchboard without human intervention; placing calls through 'virtual' operator.
- Physicians: record patient data; make records while doing observations or performing operations.
- Attorneys: use instead of secretaries; conduct online research.

1.3 Motivation of the Research

With so much convenience that speech recognition could bring to our life, there are convincing reasons for researching and improving speech recognition technology. However, achieving recognition is quite a difficult task. The complexity is due to the number of the involved speakers, the variability of utterances, the complexity of languages, and the environmental conditions under which the speech recognition system must operate.

The two main concerns in speech recognition are to improve the recognition accuracy and the processing speed. Therefore, the motivation of this research is to provide a reliable and efficient recognition method.

Before creating a general system to perform continuous recognition, this thesis deals with isolated word recognition through the use of digital processing algorithms and the application of fuzzy neural techniques. Because of the uncertainty of speech waveforms, fuzzy neural techniques are recognized as an efficient way to handle this problem. The objective of this thesis is to utilize fuzzy neural techniques in designing a speech recognition system.

1.4 Organization of the Thesis

In Chapter 2, A literature survey is reviewed on speech recognition. It gives an introduction to speech production and perception, speech signal features and fundamental speech recognition methods.

Chapter 3 describes the algorithm for speech feature extraction, which is the first step in the whole process of speech recognition. In this chapter, the LPC analysis is discussed and two different LPC-based parameters -- line spectrum frequencies and cepstral coefficients are presented for the use of speech recognition.

Chapter 4 presents the fuzzy logic and neural network theories for speech recognition. The Fuzzy c-means algorithm is introduced for clustering the word templates.

In Chapter 5, a template-based fuzzy speech recognizer is described. It also includes the recognition results and analysis.

Chapter 6 gives the conclusions and suggestions for future research.

Chapter 2

Literature Survey on Speech Recognition

2.1 Introduction to Speech Sounds

2.1.1 Speech Production

Speech sound is produced by a set of well-controlled movements of various speech apparatus. Figure 2.1 shows a schematic cross-section through the vocal tract of the apparatus.

The vocal tract is a primary acoustic tube, which is the region of the mouth cavity bounded by the vocal cords and the lips. As air is expelled from the lungs, the vocal cords are tensed and then caused to vibrate by the airflow. The frequency of oscillation is called the fundamental frequency, and it depends on the length, tension and mass of the vocal cords. During this process, the shape of the vocal tube is changed by different positions of the velum, tongue, jaw and lips [2]. The average length of the vocal tract for an adult male is about 17cm, and its cross-section area can vary in its outer section from 0 to about $20cm^2$. Therefore, the vocal tract, as an acoustic resonator, will determine variable resonant frequencies by adjusting the shape and size of the vocal tract. The resonant frequency is called the formant frequency or simply formant. The nasal tract is

an auxiliary acoustic tube that can be acoustically cooperated with vocal tract to produce nasal sounds.

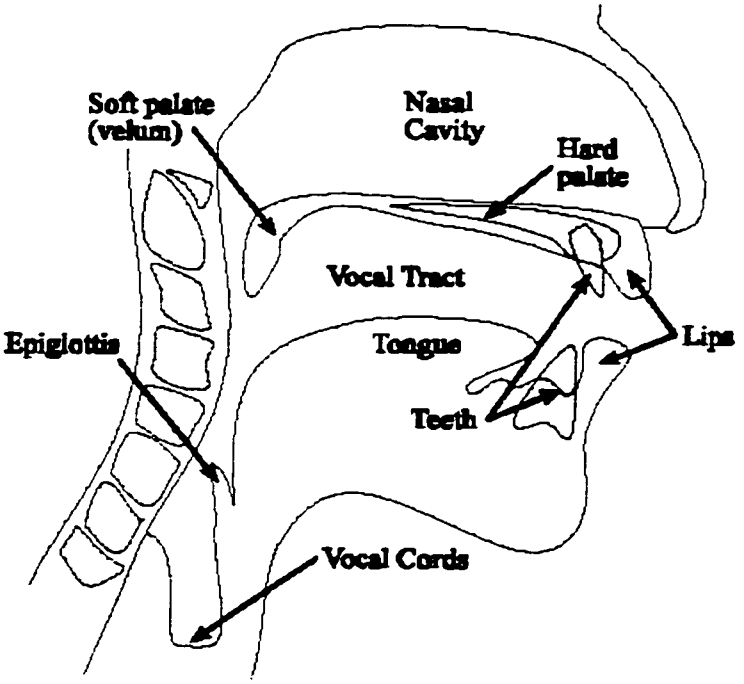


Figure 2.1: Schematic view of the human speech apparatus

Various speech sounds are produced not only by adjusting the shape of the vocal tract, but also the type of excitation. Besides the airflow from the lung, the excitation could come from some other sources: the fricative excitation, plosive excitation and whispered excitation [3].

2.1.2 Speech Perception

As the vocal system can produce speech sounds, the auditory system is capable of detecting the change in air pressure of audible sounds [2]. Figure 2.2 shows a cross-

section diagram of human ear. The ear consists of three parts: the outer ear, the middle ear, and the inner ear [26]. The outer ear collects the sound waves and passes the air pressure variations to the eardrum. The middle ear is an air-filled cavity, which serve as a mechanical amplifier and transform vibrations of the eardrum into oscillations of the fluid filled inner ear. The inner ear then converts the mechanical vibrations into electrical potentials that go to the auditory nerve and the cortex.

The human ear is most sensitive to frequencies of the range from 1000 to 4000Hz. Most speech information is covered within these frequencies. It is shown by experiments that human ears are largely phase insensitive. The basilar membrane is only deformed when the stapes pushes on the oval window [1], thus very little information is available for the brain to determine the waveform's phase. This fact could be applied to speech recognition to reduce the amount of data in the encoded waveform.

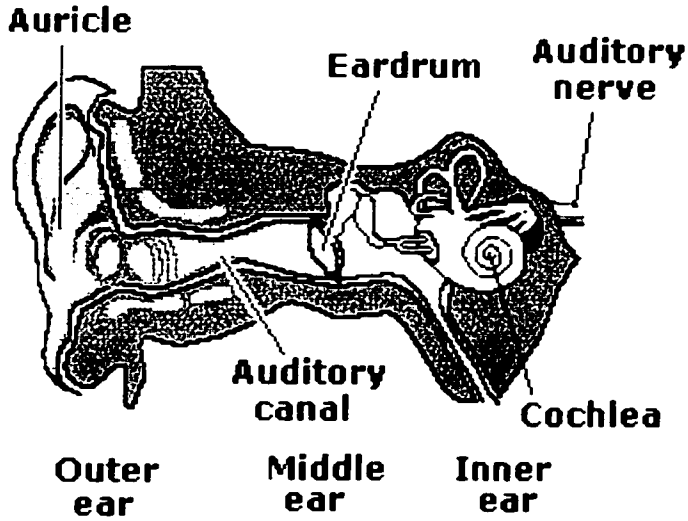


Figure 2.2: Cross-section of the human ear

2.1.3 Speech Features

The speech recognition can be divided into two processes: feature extraction and pattern recognition. Feature extraction is responsible for searching the speech characteristics and storing them for the second process: pattern recognition. In order to identify the speech characteristics accurately and efficiently, it is necessary to investigate the features and classifications of speech sounds.

Any natural language, including English, is based on a set of distinguishable and mutually exclusive primary units, which are called phonemes. All the phonemes are related to different articulatory gestures of a language.

There are several ways to classify speech sounds [1, 2]. According to the type of excitation source of phonemes, speech sounds can be classified into the following categories:

- *Voiced* sounds (/a/, /d/) occur when air pressure pushes the vocal cords open and causes them to vibrate. The vibrating cords modulate the air stream from the lungs at a rate that could be as low as 60 times per second for some males to 500 times per second for children. The peak amplitude of voiced sound is much higher than that of the unvoiced sound.

- *Nasal* sounds such as /m/, /n/ are also voiced. However, the nasal cavity is involved together with the vocal cavity during the utterance. Part of the airflow is diverted into the nasal tract by opening the velum.
- *Fricatives* are generated by exciting the vocal tract with turbulent flow created by airflow through a narrow constriction. For example, the sound /f/, /s/ and /sh/ are fricatives.
- *Voiced fricatives* occur when the vocal tract is excited simultaneously by both turbulence flow and vocal vibration. The sounds /z/, /zh/ and /v/ belong to this category.
- *Plosives* are produced by exciting the vocal tract with a rapid release of pressure by the constrictions of lips or teeth. The plosives /t/, /k/ are voiceless, while /b/, /d/ are voiced.
- *Affricative* sounds are produced by gradually releasing a completely closed and pressurized vocal tract.
- *Whispered* sounds are excited by airflow rushing through a small triangular opening between the arytenoid cartilages at the rear of the nearly closed vocal folds.

For vowel sounds, because the vocal tract remains relatively stable, three or four resonance frequencies (formants) can usually be detected from 0 to 4KHz. Therefore, the vowel sounds can be characterized by the two first formants, where the third and fourth formants are less discriminative. Table 2.1 shows the three first mean formant frequencies for eight vowels of Mid-West American English.

Table 2.1: Formant frequencies for eight vowels of Mid-West American English (After Ladefoged, 1985 [31])

Formant	/i/	/ɪ/	/ɛ/	/æ/	/a/	/ɔ/	/o/	/u/
F_1(Hz)	280	400	550	690	710	590	450	310
F_2 (Hz)	2250	1920	1770	1660	1100	880	1030	870
F_3 (Hz)	2890	2560	2490	2490	2540	2540	2380	2250

2.1.4 Representation of Speech Signal

A speech signal can be broken into several small components: phonemes, diphones, syllables or words, where a phoneme is a minimal unit of speech sound. However, it is practically difficult to identify an individual phoneme due to the overlapping of phonemes. In automatic speech recognition, isolated word is used as the minimum unit because it is relatively easier to separate it within a sentence or phase.

Speech is a slowly time varying activity which can be simply graphically displayed by its waveform. The waveform is created by air pressure controlled by the lungs, vocal tract, tongue and mouth. However, the time domain representation is much less popular than the frequency representations. This is because the human ears perform some type of frequency analysis rather than time domain analysis during the auditory process, and it is

found that the human ear is much more sensitive to the magnitude spectrum than the phase information of the speech signal.

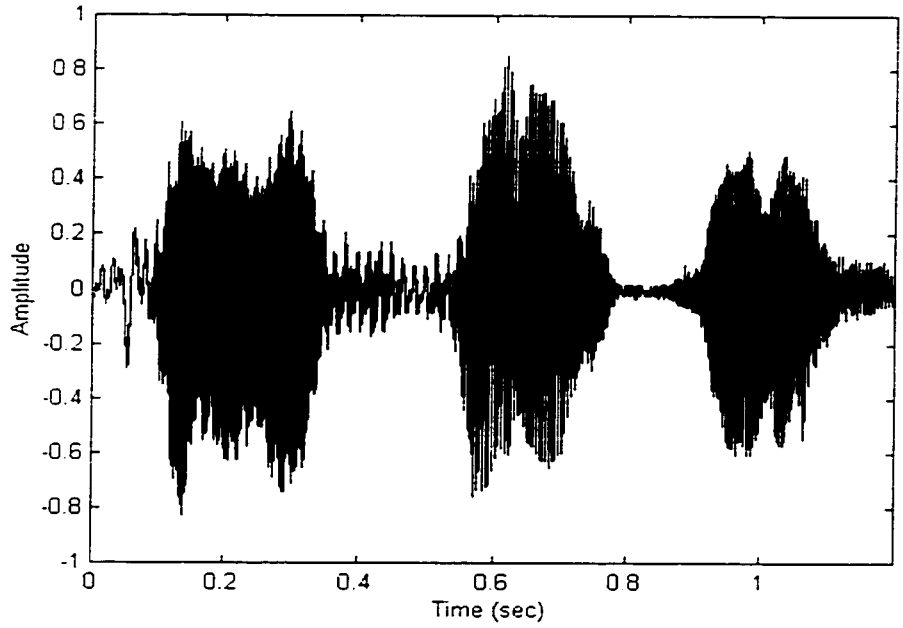


Figure 2.3: Waveform of the sentence "please log in"

The most popular representation of a speech signal is the spectrogram, which is a three-dimensional representation on the time-frequency domain. The introduction of the spectrogram provided a way to produce a display of the time varying spectral characteristics of speech. An example of spectrogram is shown in Figure 2.4. The vertical axis represents frequency while the horizontal corresponding to time. The darkness shows the signal energy at a certain time and frequency, and the location of dark areas change while the pronunciations move from one vowel to another inside the

utterance. Therefore, the formant frequencies of the vocal tract show up as dark bands in the diagram. For example, the first two dark bands in "please" are located around 300Hz and 2200Hz: while they are 600Hz and 1000Hz in word "log".

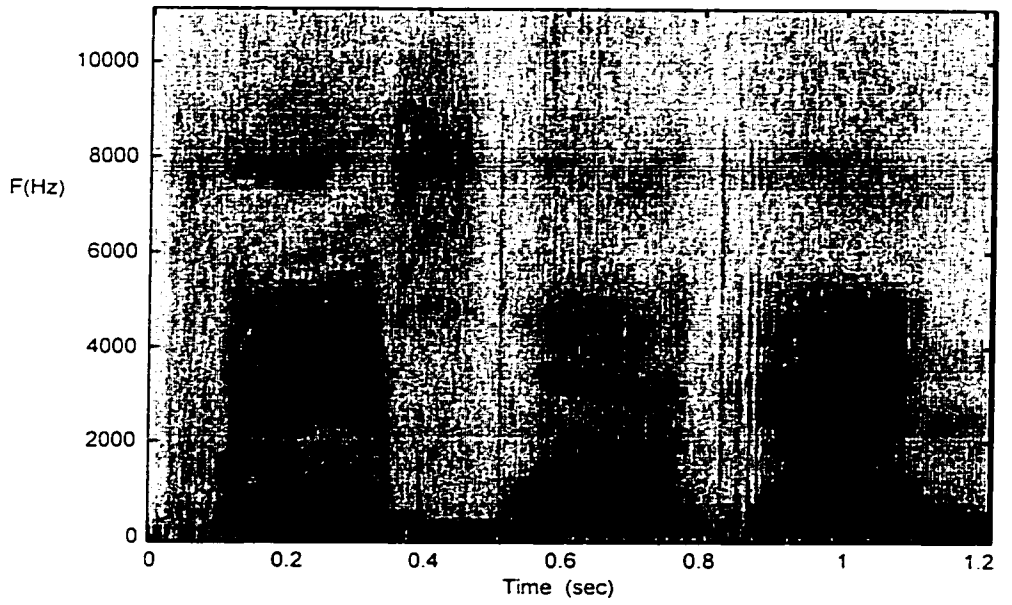


Figure 2.4: Spectrogram of the sentence "please log in"

Generally, voiced regions are featured by a striated appearance due to the periodicity of the waveform, while unvoiced regions are more evenly filled in. This phenomenon is shown in Figure 2.5, which gives the waveform and spectrogram of a sentence "starting to download". It is obvious that there are dark bands for voiced region and lighter color is distributed for unvoiced regions. This is in coincidence with the fact that only the voiced sounds have formant frequencies.

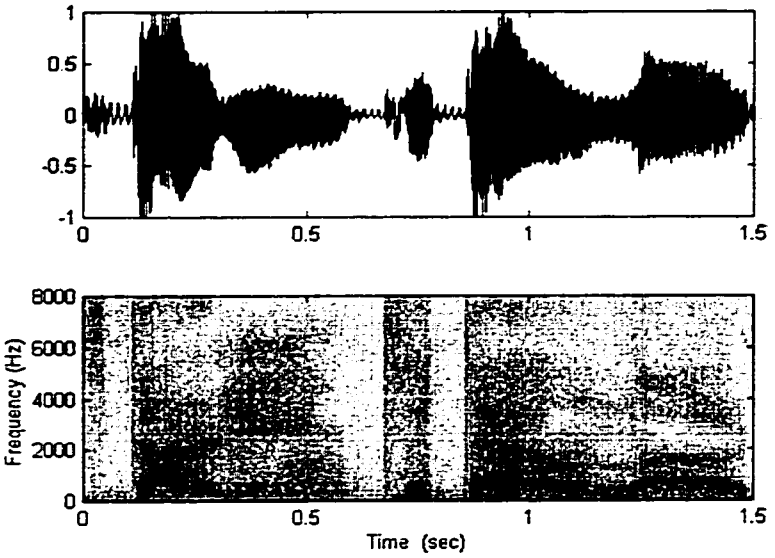


Figure 2.5: Waveform and spectrogram of the sentence "starting to download"

2.2 Fundamental Speech Recognition Techniques

2.2.1 Classification of Speech Recognition

Automatic speech recognition can be classified into a number of different categories depending on different issues:

1. The manner in which a user speaks. Usually there are three recognition modes based on the speaking manner:
    - Isolated word recognition: The user speaks individual words or phrases from a specified vocabulary. Isolated word recognition is suitable for command recognition.
    - Connected word recognition: The user speaks fluent sequence of words with small spaces between words, in which each word is from a specified vocabulary (e.g., zip codes, phone numbers).
    - Continuous speech recognition: The speaker can speak fluently with a large vocabulary.
2. The number of users:
    - Speaker dependent: The users of a recognition system only consist of a single speaker or a set of known speakers.
    - Speaker independent: arbitrary users will use the ASR system in this case.
    - Speaker adaptive: The system will customize its response to each individual speaker while it is in use by the speaker.
3. The size of the recognition vocabulary:

- A small vocabulary system only provides recognition capability for a small amount of words.
- A large vocabulary system is capable of recognizing words among a vocabulary containing up to 1000 words.

4. The degree of dialogue between the human and the machine, including:
   - One-way communication in which each user spoken unit is acted upon.
   - System driven dialog systems in which the system is the only initiator of a dialog, requesting information from the user via verbal input.
   - Natural dialogue systems in which the machine conducts a conversation with the speaker, solicits inputs, acts in response to user inputs, or even tries to clarify ambiguity in the conversation.

2.2.2 *Difficulties in Speech Recognition*

Because speech waveform is nonlinear and dynamic, speech recognition is an inherently difficult task. There are several main variabilities of speech signal including within-speaker variability, across-speaker variability, transducer and transmission variability, language complexity, and the environmental conditions under which a speaker is talking.

Within-speaker variability is caused by inconsistent pronunciation, speaking speed and different emotions when the words or phrases are spoken by same speaker.

Across-speaker variability is due to the physiological differences, regional accents, foreign languages, etc. The physiological correlates are associated with the size and configuration of the components of the vocal tract of each individual. The variations in the vocal tract can cause different resonance frequencies (formants) and pitch frequency of the same words.

Transducer and transmission variability is because the words are spoken over different microphone/handsets and the speech signal could be transmitted by all kinds of communication systems (telecommunication networks, cellular phones, etc.), in which unexpected noises are introduced into the signal.

Language complexity makes speech recognition an extremely difficult job. So far, the task of speech recognizers is simplified by limiting the number of possible utterances by the imposition of semantic constraints. On the other hand, we shall obey multi-disciplinary natures of speech signal and be adaptive to the language complexity because speech is a completely natural activity of human beings.

Environmental condition is also a main concern of speech recognizers while real applications usually are conducted in adverse conditions which may drastically degrade the system performance. Therefore, it is necessary to present robust recognition methods for dealing with reasonable noise or distortions of the speech signal.

2.2.3 Speech Recognition Approaches

2.2.3.1 Acoustic-Phonetic Approach

The earliest approaches of speech recognition were based on the theory of acoustic phonetics to find speech sounds and provide phonetic characteristic labels for these sounds. These existing finite, distinctive phonetic units in spoken language could be characterized by a set of acoustic properties which are manifest in the speech signal over time. The first step in the acoustic-phonetic approach is to segment the speech signal into stable acoustic regions and label them, followed by adding one or more phonetic labels. The second step is to determine a valid word from the phonetic label sequences based on the first step. Because the difficulty of getting a reliable phoneme lattice in step one, the acoustic-phonetic approach has not been widely used for most commercial applications.

2.2.3.2 Pattern Matching Approach

The pattern matching approach is based on pattern recognition algorithms that require pattern templates before recognition [2]. It has two steps: Pattern training and pattern comparison (Figure 2.6). Pattern training is responsible for establishing consistent speech pattern representation for a set of known training samples. There are several methods for training such as statistical models (e.g., hidden Markov model) and clustering training (learning vector quantization, fuzzy c-mean clustering). The second

step. pattern comparison compares the unknown speech with each template, and determine the identity of it by the matching algorithms.

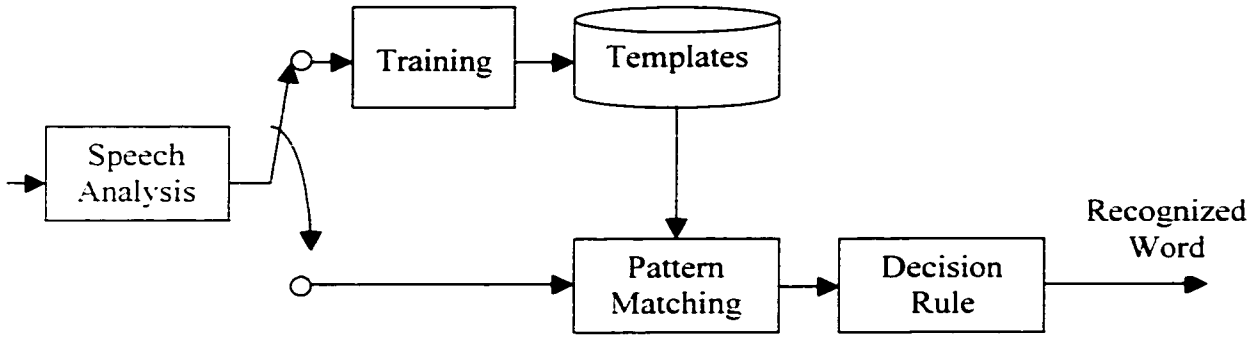


Figure 2.6: Block Diagram of Pattern Recognition Recognizer

2.2.3.3 Computational Intelligence approach

The computational intelligence approach is a hybrid method of the acoustic-phonetic approach and the pattern matching approach. Generally a neural network is applied to integrate the knowledge of speech for segmentation and labeling, and intelligent tools are used for learning the relationship among phonetic events. This method has been proved to be a very promising area for speech recognition and was widely used in commercial applications.

Chapter 3

Speech Feature Extraction

3.1 Linear Predictive Analysis

Linear predictive analysis has been one of the most powerful speech analysis techniques since it was introduced in the early 1970s. Primarily it is a time-domain coding method for low bit rate speech storage and transmission, but it can also be used for providing frequency-domain parameters (like formant frequency, bandwidth etc.) on the time basis of the speech signal. In the application of speech recognition, these parameters can serve as the speech characteristics representation.

For speech recognition, linear predictive coding (LPC) has several advantages over other techniques including:

- LPC is capable of providing accurate estimates for the speech spectrum envelope. It can be used to separate the excitation source properties of pitch and amplitude from the vocal tract filter which controls the phoneme articulation and is directly related to the produced speech sounds.
- LPC is easy to be implemented by either software or hardware because it is mathematically precise, simple and straightforward.

- The LPC algorithm is computationally efficient. The required amount of computation of LPC is much less than that of other techniques such as the fast Fourier transform or filter bank model.

3.1.1 The LPC Model

The LPC is a model based on the vocal tract of human beings [4]. The basic idea of LPC model is that a speech sample *x(n)* can be predicted by a linear combination of several past sample values of speech:

$$x'(n) = a_1 x(n-1) + a_2 x(n-2) + ... + a_p x(n-p) \qquad (3.1)$$

Where $a_1, a_2, ..., a_p$ are called linear predictive coefficients, and they should be optimized to minimize the prediction error between the actual signal and the predicted values of this sample, which is:

$$\begin{aligned} e(n) &= x(n) - x'(n) \\ &= x(n) - a_1 x(n-1) - a_2 x(n-2) - ... - a_p x(n-p) \\ &= x(n) - \sum_{k=1}^{p} a_k x(n-k) \end{aligned} \qquad (3.2)$$

Although the speech signal is nonlinear and quite variant, the speech waveform over a short periods of time (around 10 to 30 msec) still remains roughly invariant. Therefore, the LPC coefficients can be re-calculated to minimize the mean squared prediction error

over a short frame of the speech waveform, with each frame segmented to a length of around 10 to 30 msec.

Transforming the predictive error in equation 3.2 from time domain into z-transform gives:

$$E(z) = X(z) - \sum_{k=1}^{p} a_k z^{-k} X(z) = \left(1 - \sum_{k=1}^{p} a_k z^{-k}\right) X(z) \qquad (3.3)$$

Therefore, the transfer function between the speech sample and the prediction error could be written as:

$$H(z) = \frac{X(z)}{E(z)} = \frac{1}{\left(1 - \sum_{k=1}^{p} a_k z^{-k}\right)} = \frac{1}{A(z)} \qquad (3.4)$$

When the LPC model is applied to a speech signal, the predictive error E(z) can be identified as the impulsive excitation of the vocal tract, while the all pole system H(z) represents the vocal tract model. This is how linear prediction separates out the excitation properties of the source from the vocal tract filter: the source parameters are derived from the prediction error, and the vocal tract filter is characterized by the linear predictive coefficients. Based on the analysis experiments, the excitation source is essentially a quasi-periodic pulse train for voiced speech signals, and a random noise signal for unvoiced sounds.

A speech synthesis model is built in Figure 3.1 based on the LPC model. The normalized excitation signal *u(n)* is set to be either a quasi-periodic impulse train or a random noise

signal (depending on the voiced/unvoiced determination). The appropriate gain of the source G is estimated from the signal, and the scaled source is fed as input to a digital filter that represents the vocal tract model.

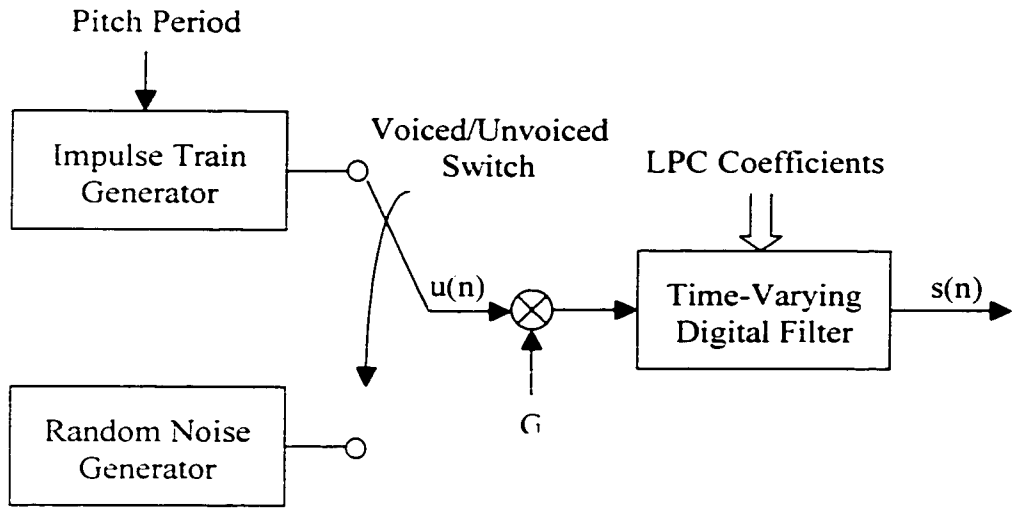


Figure 3.1 Block diagram of LPC-based speech synthesis model

There are three basic algorithms to compute the LPC coefficients which could minimize the prediction error over a speech frame [4]:

- The autocorrelation method
- The covariance method
- The lattice method

Among these three methods, the autocorrelation method is the most common used method for linear predictive analysis. Defining the autocorrelation coefficients of speech samples are given by:

$$R_n = \sum_{i=1}^{N-n} x(i)x(i+n) \tag{3.5}$$

Then the linear prediction coefficients can be computed using the Durbin-Levinson's recursive algorithm as shown below [2]:

$$\begin{aligned} E_0 &= R_0 \\ k_i &= \frac{R_i - \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R_{i-j}}{E_{i-1}} \qquad for \qquad 1 \le i \le p \\ \alpha_i^{(i)} &= k_i \\ \alpha_j^{(i)} &= \alpha_j^{(i-1)} - k_i \alpha_{i-j}^{(i-1)} \\ E^{(i)} &= (1 - k^2) E^{(i-1)} \end{aligned} \tag{3.6}$$

where the final solution of LPC coefficients are given as $a_m = \alpha_m^{(p)}$, for $1 \le m \le p$.

In speech recognition, the LPC model is used as a characteristic model for a speech signal. Figure 3.2 and 3.3 gives the comparison between the original speech power spectrum and the magnitude spectrum of the LPC model. It's obvious that LPC provides a good approximation to the vocal tract spectral envelope, in which the information of formant frequency and magnitude are included for speech recognition.

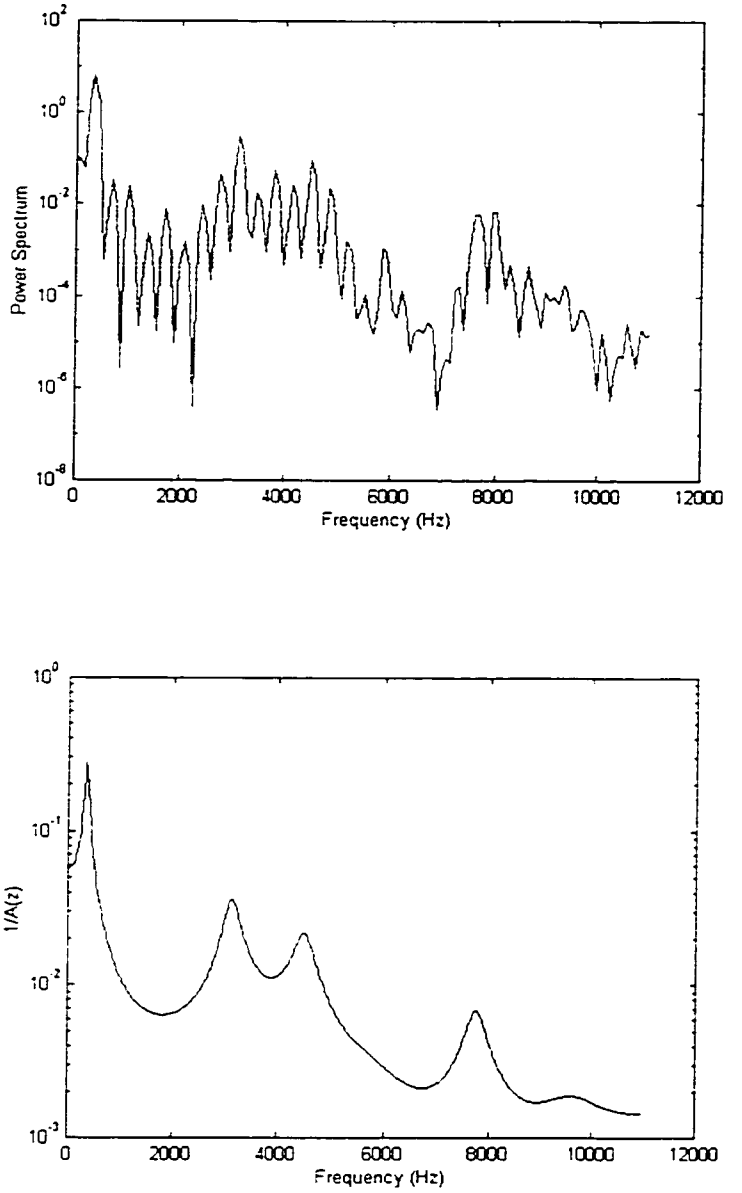


Figure 3.2 (a) Original power spectrum (b) Magnitude of LPC model for phoneme /i/

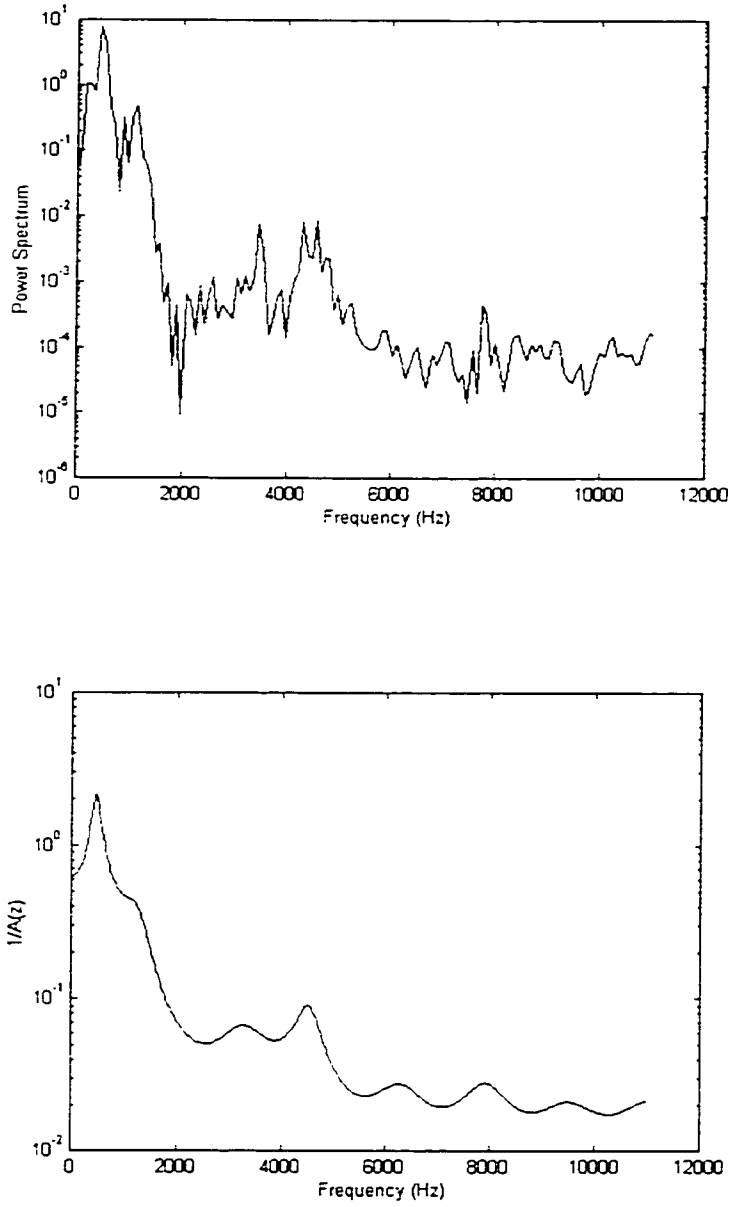


Figure 3.3 (a) Original power spectrum (b) Magnitude of LPC model for phoneme /o/

3.1.2 *LPC Processor for Speech Recognition*

The LPC technique is used to build a front-end processor for a speech recognition system to process a speech signal, s(n), as shown in Figure 3.4.

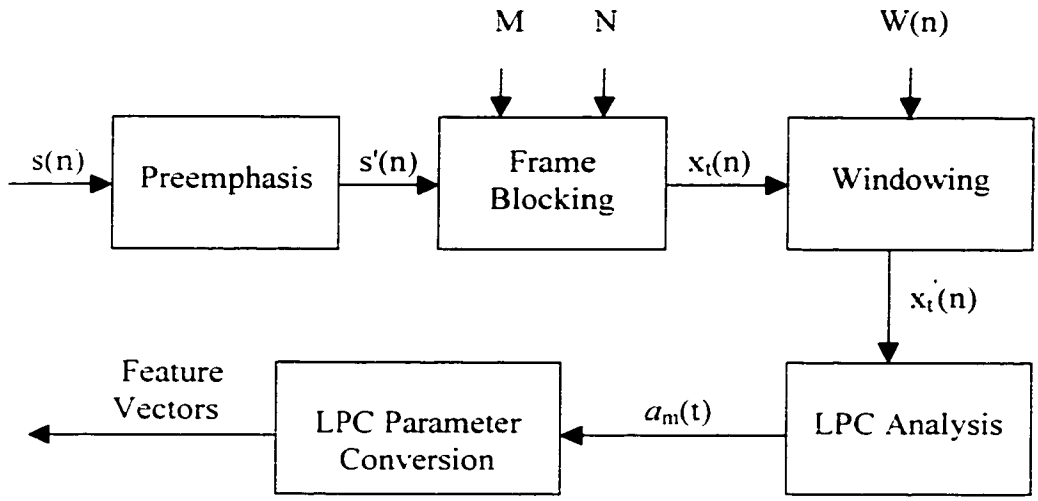


Figure 3.4 Block diagram of LPC processor

The LPC processor includes following basic steps:

1. *Preemphasis:* A low-order system is applied to the speech signal in order to spectrally flatten the signal and to make it less susceptible to finite precision effects for the signal processing. The most widely used preemphasis filter is a first order system:

$$H(z) = 1 - az^{-1}, \qquad 0.85 \le a \le 1 \tag{3.7}$$

where the parameter a is usually set to be 0.95. After applying this filter, the output s'(n) and input s(n) have the following relationship in the time domain:

$$s'(n) = s(n) - as(n-1) \tag{3.8}$$

2. *Frame Blocking:* The preemphasized speech signal s'(n) is segmented into small frames, with N samples for each frame. Between the adjacent frames, there's M samples overlapping to prevent the spectral discontinuous after blocking.

3. *Windowing:* After blocking the frames, a window is applied to each frame to minimize the spectral discontinuities at the beginning and the end of the speech frame:

$$x_t'(n) = w(n)x_t(n), \qquad 0 \le n \le N-1 \tag{3.9}$$

A typical window is the Hamming window:

$$w(n) = 0.54 - 0.46\cos(\frac{2\pi n}{N-1}), \qquad 0 \le n \le N-1 \tag{3.10}$$

4. *LPC analysis:* For each frame, the LPC coefficients are calculated according to the recursive equation 3.6.

5. *LPC parameter conversion:* In general, direct quantization and application of LPC coefficients is inefficient and unreliable because the LPC coefficients are too dynamic and a small quantization error could cause the entire filter to be unstable and

inaccurate. Due to this weakness of LPC coefficients, some other related coefficients are considered, such as the reflection coefficients, cepstral coefficients and line spectral frequencies (LSFs). In this thesis, line spectral frequency and cepstral coefficients are used as the extracted speech features. These two parameters are described in the following section.

3.2 Line Spectrum Frequency

The line spectrum frequency was first proposed by Itakura in 1975 [6] as an alternative parametric representation for the LPC model. In the context of speech coding, LSF has been shown to have better quantization and interpolation properties than other representations such as reflection coefficient and log area ratio of the LPC model. Also, a number of researchers have shown that a speech recognition system can benefit from these advantages of LSF [7, 8, 9, 10].

Algorithm and Properties of LSF

In the LPC analysis of speech, assuming a speech frame is modeled by an all-pole filter $H(z) = 1/A(z)$ with order p, where $A(z)$ is the inverse filter given by:

$$A(z) = 1 + a_1 z^{-1} + \ldots + a_p z^{-p} \qquad (3.11)$$

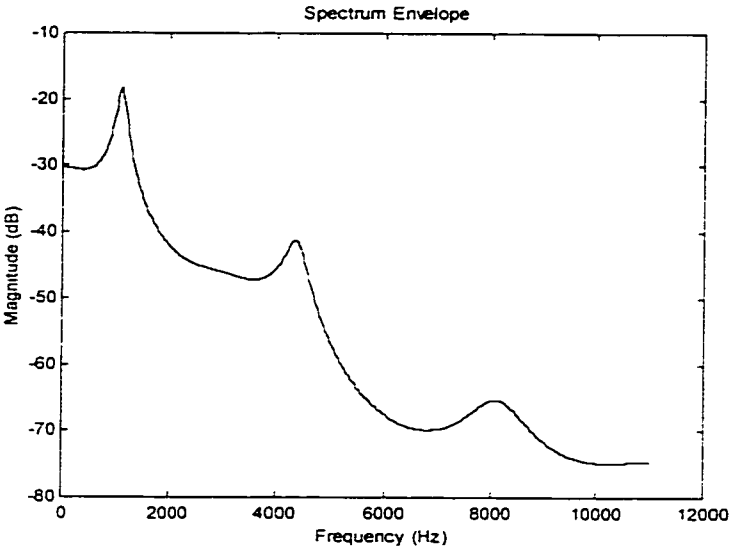
The LSF is a represented by mapping the p zeros of A(z) onto the unit circle through a pair of (p+1) order polynomials P(z) and Q(z):

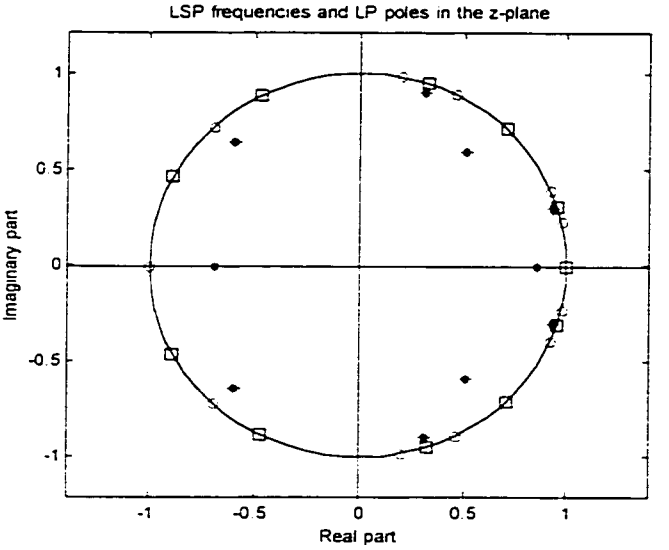
$$P(z) = A(z) + z^{-(p+1)} A(z^{-1}) \quad (3.12)$$

$$Q(z) = A(z) - z^{-(p+1)} A(z^{-1}) \quad (3.13)$$

These polynomials can be shown to have some interesting properties. The first is that all the zeros of P(z) and Q(z) lie on the unit circle and they are interlaced with each other. Secondly, the frequencies tend to be clustered near the format frequencies; when the P(z) and Q(z) frequencies are close, it is likely that the original A(z) zero was close to the unit circle, and a formant frequency is likely to be located between the corresponding frequency pair. Also, the closer one pair is, the sharper the formant will be. Thus, the LSF could be utilized as the frequency features for speech recognition systems. Figure 3.5 shows the spectrum and LSF of a speech segment, which demonstrates the above two properties. Figure 3.6 and 3.7 give the LSF plots of two isolated words.

LSF have attracted much interest because they are good representations of LP systems, and typically result in quantizers having either better representation or using fewer bits for equivalent representation than reflection coefficient quantizers.



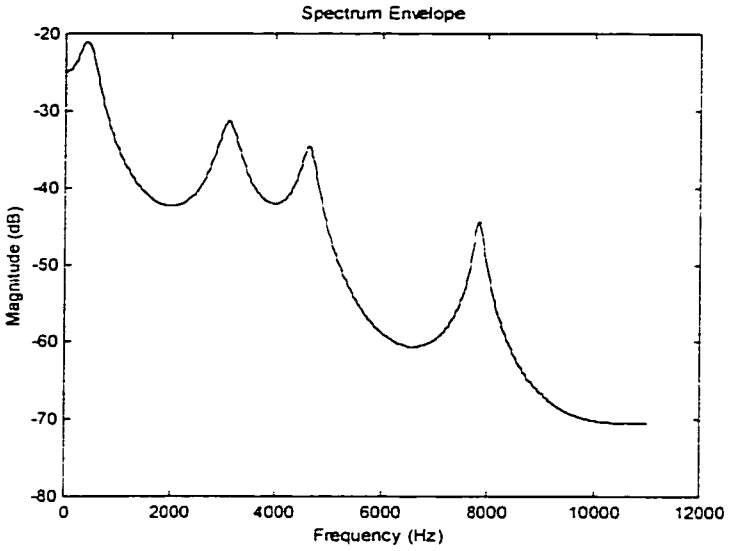


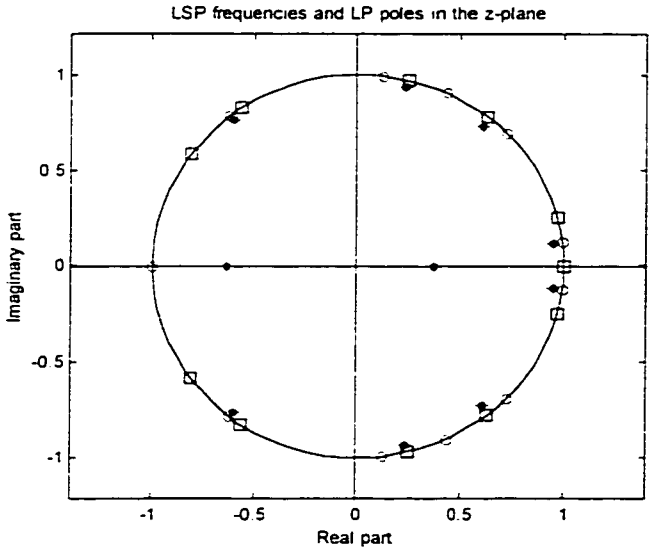
*: LP poles

O: P(z) zeros

□: Q(z) zeros

Figure 3.5 LSF and LP poles in the z plane of phome /o/





*: LP poles

O: P(z) zeros

□: Q(z) zeros

Figure 3.6 LSF and LP poles in the z plane of phonme /i/

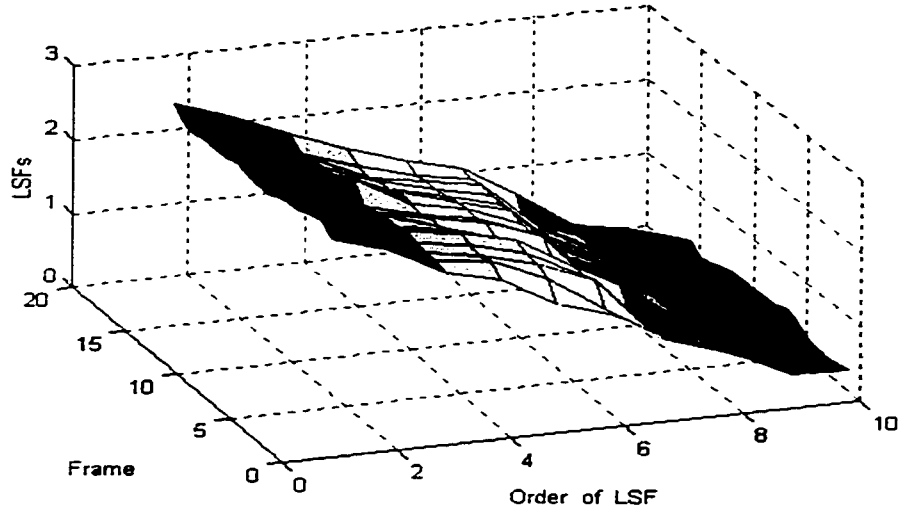


Figure 3.7 LSF of word "no"

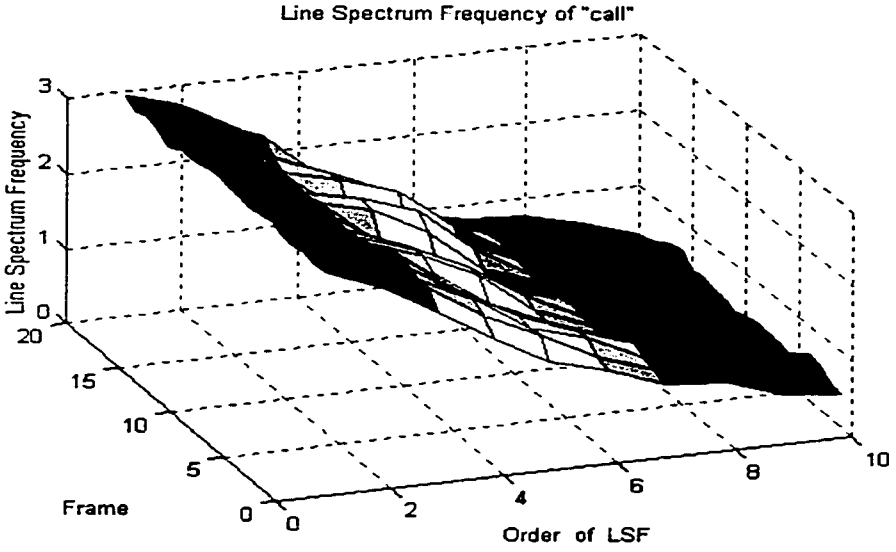


Figure 3.8 LSF of word "call"

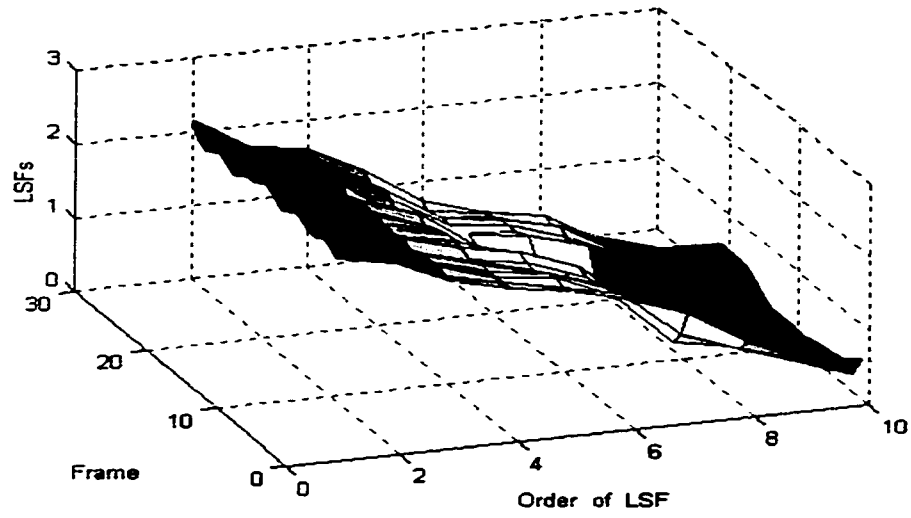


Figure 3.9 LSF of word "hangup"

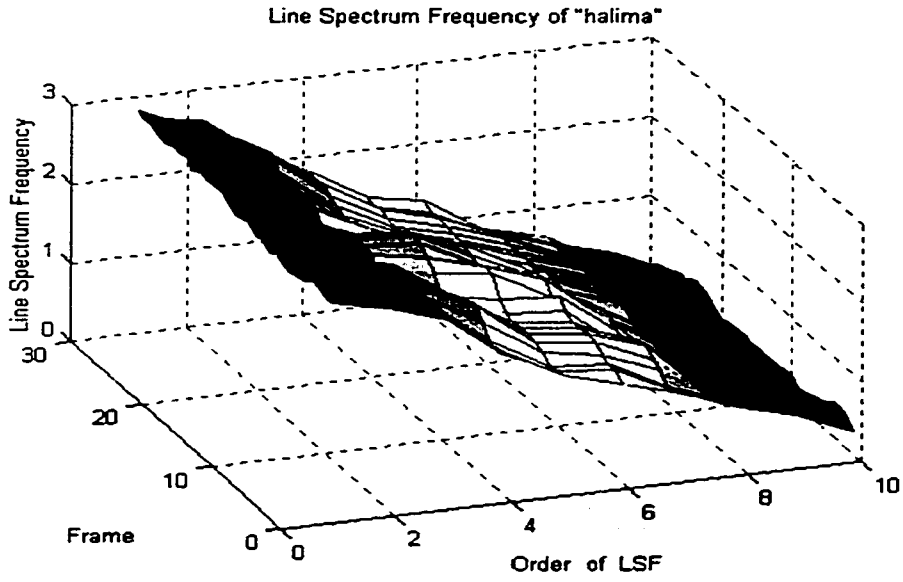


Figure 3.10 LSF of word "Halima"

Cepstral Coefficients

Cepstral coefficients have been proved to be another efficient and robust feature set for speech recognition. Originally, the cepstrum of a speech signal $x(n)$ is defined as the Fourier transform of the logarithm of the magnitude of the spectrum $X(e^{jw})$:

$$c_m = \frac{1}{2\pi}\int_{-\pi} \log\left|X(e^{jw})\right| e^{jwm}\, dw \tag{3.14}$$

Based on the LPC model, by applying the smoothed magnitude as the magnitude spectrum, cepstral coefficients can be derived directly from the LPC coefficient set with the recursive formula:

$$\begin{aligned} c_0 &= \ln G \\ c_m &= a_m + \frac{1}{m}\sum_{k=1}^{m-1} k c_k a_{m-k}, \qquad 1 \le m \le p \\ c_m &= \frac{1}{m}\sum_{k=1}^{m-1} k c_k a_{m-k}, \qquad m > p \end{aligned} \tag{3.15}$$

Properties of Cepstral Coefficients

To make use of the cepstral coefficients properly for speech recognition, it's necessary to know the properties of them:

- Most information of speech signal is represented by the lower numbered cepstral coefficients, and the first p coefficients can uniquely determine the all-pole filter of LPC model

- Cepstrum is a decaying sequence, under regular conditions, the variances of coefficients (except c_0) are essentially inversely proportional to the square of the coefficient index (Figure 3.11 - 3.14)

Because the cepstrum has infinite index numbers, only the first 10 to 30 coefficients are taken for representing the speech feature based on the above properties of cepstral coefficients.

Weighted Cepstral Coefficients

The variance of cepstral coefficients is inversely proportional to the square of the coefficient index as follows [2]:

$$E(c_m^2) \propto \frac{1}{m^2} \qquad (3.16)$$

The cepstral coefficients can be normalized with the index m, to balance the contribution from each cepstral coefficient, then the weighted coefficients become:

$$\hat{c}_m = m c_m, \qquad 1 \le m \le L \qquad (3.17)$$

A more complicated weighting function can be applied for de-emphasiszing the coefficient around *m=1 and l*:

$$\hat{c}_m = \left[1 + \frac{L}{2}\sin(\frac{n\pi}{L})\right] c_m, \qquad 1 \le m \le L \qquad (3.18)$$

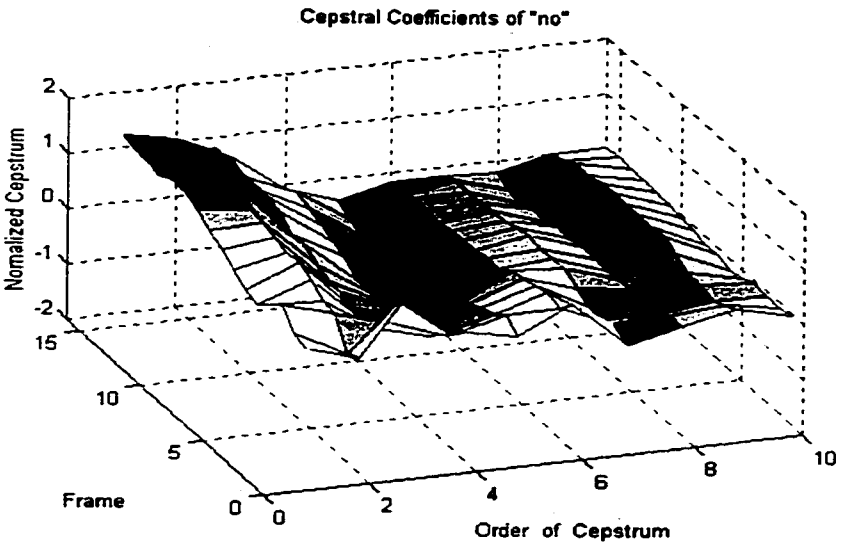


Figure 3.11 Cepstral coefficients of word "no"

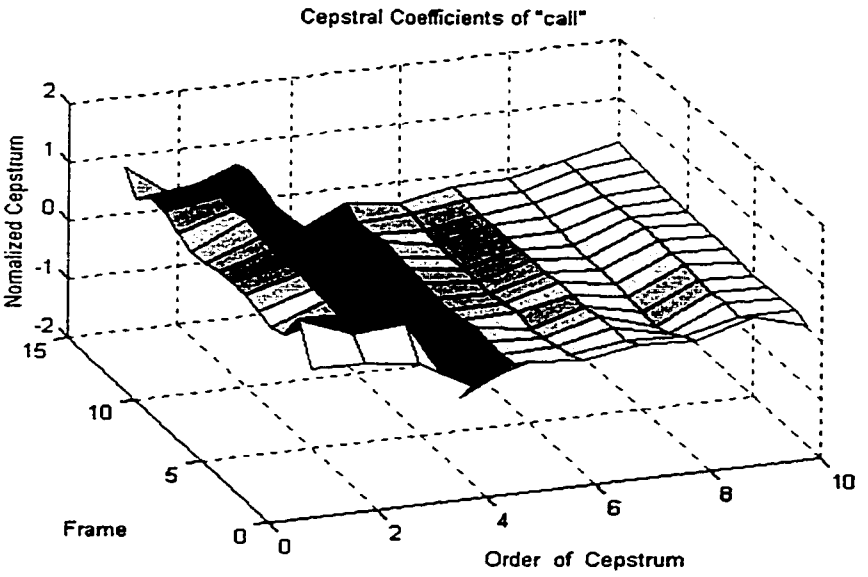


Figure 3.12 Cepstral coefficients of word "call"

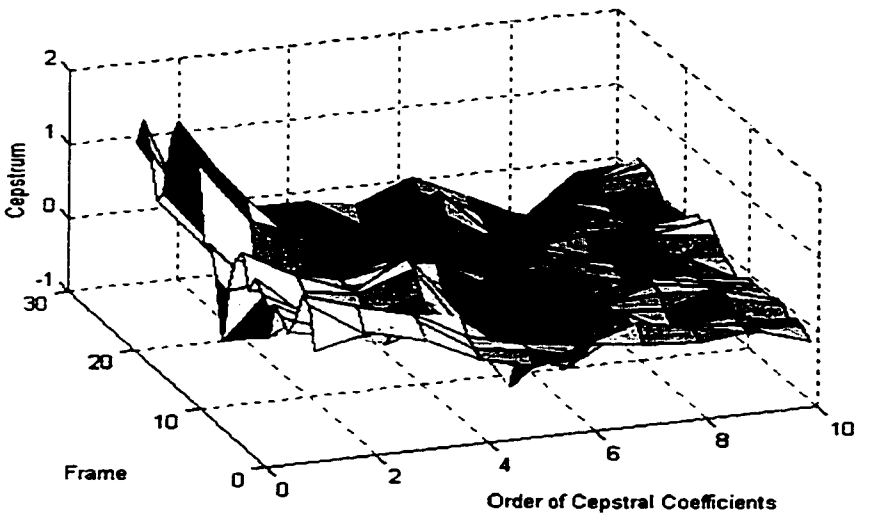


Figure 3.13 Cepstral coefficients of word "hangup"

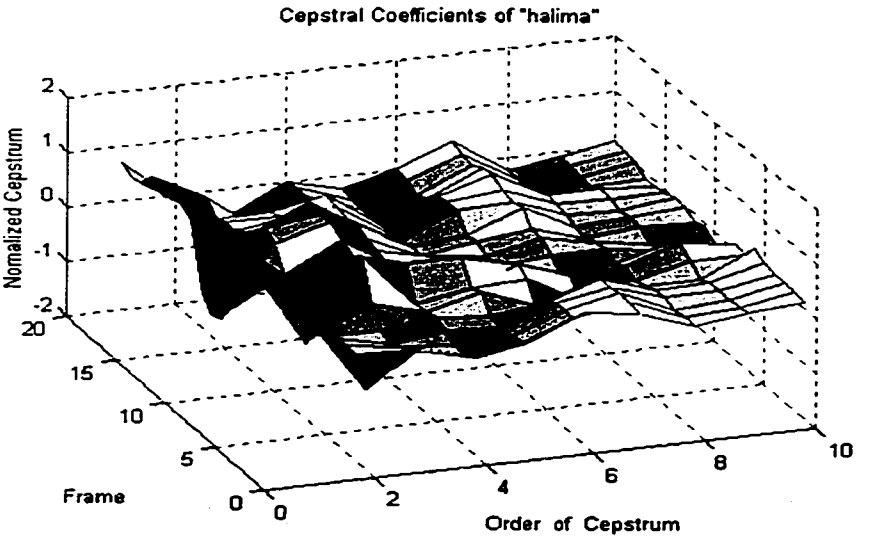


Figure 3.14 Cepstral coefficients of word "halima"

Chapter 4

Fuzzy Neural Network for Speech Recognition

4.1 Fuzzy Logic

4.1.1 Background

Fuzzy sets were introduced by Zadeh [32] in 1965 as a new way to represent and manipulate data with uncertainty and fuzziness. In the old paradigm, fuzziness was considered unfavorable because of the expectation for scientific precision and accuracy. However, fuzzy interpretations of data is a natural and intuitively plausible way to formulate and solve a lot of problems in our everyday life. For example, expressions with uncertainty like "hot coffee", "heavy objects", and "warm weather" are fuzzy interpretations.

Although both fuzzy sets and statistical theory can deal with uncertainty, fuzzy sets are quite different from statistical models in some ways. Probabilities represent the likelihood of a certain event with a distribution among all the events, while a fuzzy set represents the applicability of the element to the set. In another word, the fuzziness provides more uncertainty that can be found in the meanings of many words from human's thinking.

Today, we have witnessed a rapid growth in a variety of applications of fuzzy logic. The applications range from consumer products such as washing machines, cameras, camcorders, and microwave ovens to industrial process control, medical instrumentation, pattern recognition, decision-support systems, and portfolio selection. As we know, communication by speech is a natural activity of human beings and contains a lot of uncertainty during both the speech production and recognition process. The application of fuzzy logic to speech recognition actually simulates the way that people understand each other every day. The reasons why fuzzy logic can be applied to speech recognition are described as following:

- Fuzzy logic is conceptually easy to understand. The mathematical concepts behind fuzzy reasoning are very simple. What makes fuzzy attractive is the "naturalness" of its approach and not its far-reaching complexity.
- Fuzzy logic is flexible with tolerance for imprecise data. Everything is imprecise if you look closely enough, but more than that, most things are imprecise even under careful inspection.
- Fuzzy logic can model nonlinear functions of arbitrary complexity. A fuzzy system can be created to match any set of input-output data. This process is made particularly easy by adaptive techniques like ANFIS (Adaptive Neuro-Fuzzy Inference Systems).
- Fuzzy logic is based on natural language. The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic. Natural language, which is used by ordinary people on a daily

basis, has been shaped by thousands of years of human history to be convenient and efficient. Sentences written in ordinary language represent a triumph of efficient communication.

4.1.2 Fuzzy sets and Fuzzy Logic

Fuzzy sets are a super-set of classical sets. In a fuzzy set, each element is associated with a real value which represents the degree of membership of the element in the closed unit interval [0, 1]. However, in classical crisp sets, all element can only be classified as "0" or "1". When all elements in a set have either complete membership or complete non-membership, the fuzzy set reduces to a crisp set.

Suppose a fuzzy set A is a subset in space X which admits partial membership. It is defined as the ordered pair $A = \{x, m_A(x)\}$, where $x \in X$ and $0 \leq m_A(x) \leq 1$. Every fuzzy set consists of the three parts: a horizontal axis x specifying the population of sets; a vertical membership axis $m_A(x)$ which specifies the membership degree of each element; and the surface itself to provide a one to one connection between the elements and their corresponding membership degree.

For example, let fuzzy set X represent the concept of "tall" for women over 20. Women 5 feet or less than 5 feet have no membership in the set "tall", while women over 6 feet have total membership. To determine the membership for a specific height, the height is

first found on the horizontal axis, then following the membership degree function, the value of membership will be located from the vertical axis. Figure 4.1 illustrates this example for fuzzy set "tall", while heights between 5 feet and 6 feet are proportionally distributed.

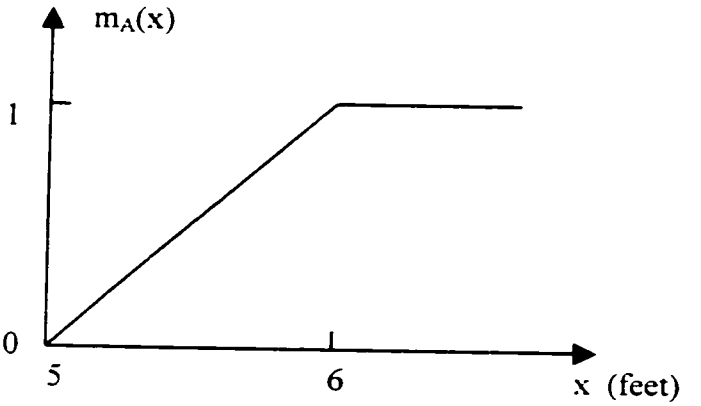


Figure 4.1 Membership function of fuzzy set for the concept "tall"

The ideal fuzzy sets representing a concept could be further expanded by linguistic variables. A linguistic variable is assigned to a fuzzy region consisting of a set of fuzzy sets. Figure 4.2 shows an example expanded from the example in Figure 4.1 for the concept "height". The variable consists of three fuzzy sets: short, medium and tall. The horizontal axis specifies the base variable of height, and the degree of membership in each fuzzy set are determined by the vertical axis.

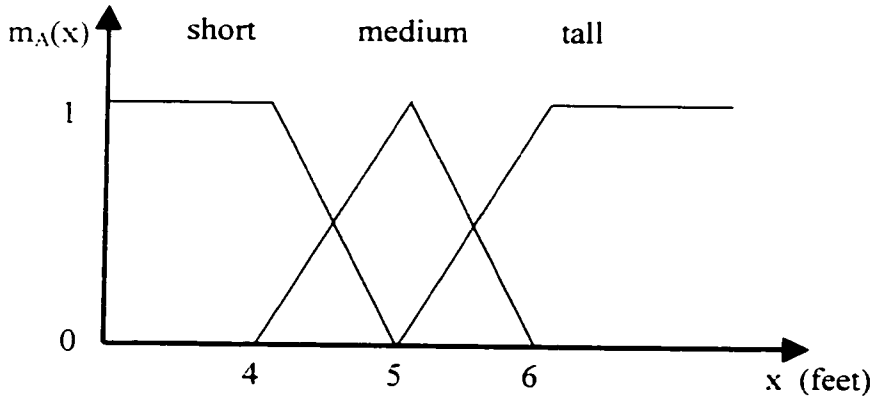


Figure 4.2 A linguistic variable of "height"

4.1.3 Fuzzy system

Fuzzy systems use fuzzy set theory to deal with fuzzy or non-fuzzy information. Generally, a fuzzy system consists of a fuzzification subsystem, a fuzzy inference engine, a fuzzy rule base and a defuzzifier as shown in Figure 4.3. The fuzzy rule base and fuzzy inference engine is the core of the fuzzy-rule-based system. A fuzzy rule can be expressed by a set of fuzzy inference rules in the form of "IF x is A THEN y is B" [19], [20]. The inference engine then implements a fuzzy inference algorithm to determine the fuzzy output from the inference rules and the inputs.

Note that a given input may simultaneously be a member of more than one set within a single fuzzy region. The inference engine interacts with the rule base and uses the inputs to determine which rules are applicable. The outputs are a set of fuzzy sets defined on the universe of possible outputs which will be defuzzified to generate crisp outputs.

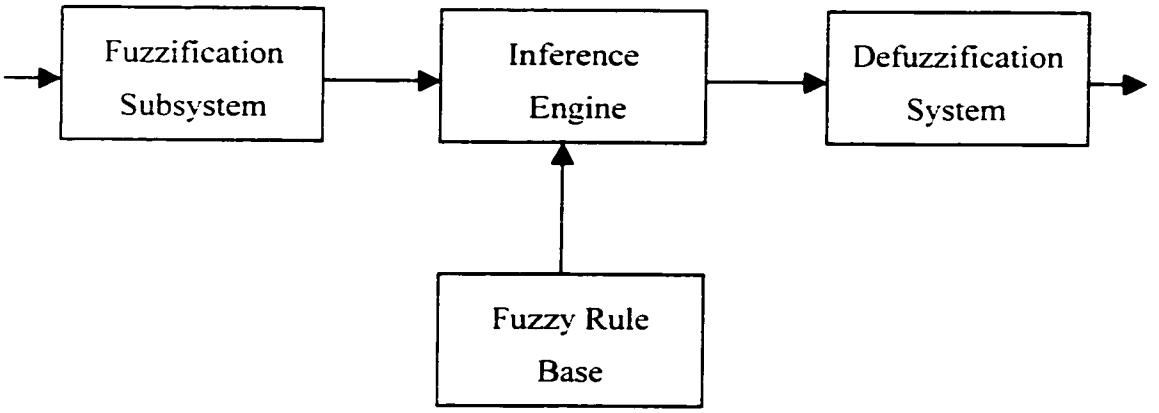


Figure 4.3 A typical fuzzy rule based system

4.2 Fuzzy Neural Networks

4.2.1 Neural network for Speech Recognition

Traditional methods for speech recognition include Hidden Markov Models (HMM) and Dynamic Time Warping (DTW). HMM is a stochastic based approach, representing the system with a number of states and calculating the probability to move from one state to another depending on the input to the system. DTW adjusts the test pattern to conform more closely with a number of templates with dynamic algorithms. Recently, Artificial Neural Networks (ANNs) have become more and more popular for speech recognition.

Artificial neural networks were first proposed in the 1940s. However, interest in this field was increased in the early 1980s. The advantages of neural networks include: massively parallel processing with high speed, robustness to complicated environments, learning ability, fault tolerance and the ability to process incomplete data. All of these make neural networks a very powerful approach for processing speech information. Neural network methods are also referred to as parallel distributed processing or connectionist approaches.

The discipline of neural networks has grown rapidly in recent years. Many researchers have successfully presented and applied neural network in many fields such as speech

recognition, image pattern recognition, sonar and radar signal processing and adaptive control systems [5].

The use of neural network models are motivated by models of neural systems of living organisms, which are composed of large number of neurons and act in a very complicated way. The basic processing unit of a neural system is called neuron (Figure 4.4). A neuron consists of three parts:

- Dendrite: receive impulses from other neurons
- Cell body (Soma): receive series of impulses and results in increasing probability that an impulse will be triggered by the cell
- Axon: Carry the impulses from cell body to next neuron

Figure 4.5 gives an example of a typical processing unit for an artificial neural network. Each neuron has a number of inputs and an output. Similarly to a neuron of a living organism, the processing unit receives the multiple inputs and after performing certain functions f, it sends out the calculated result as output, like a natural neuron being triggered by input impulses. The output of a neuron may be passed to other neurons, or recorded as one of the outputs of the system.

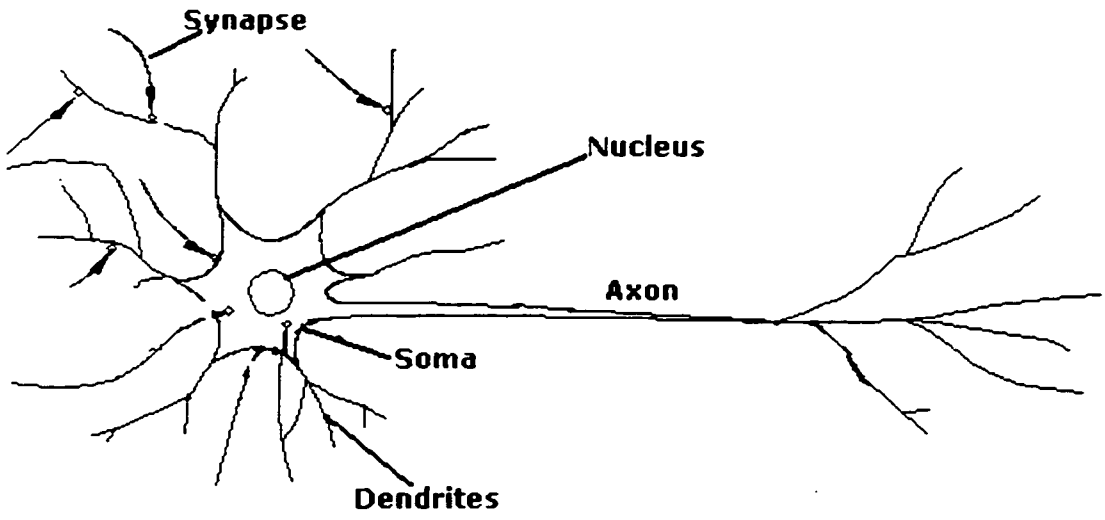


Figure 4.4 A biological neuron

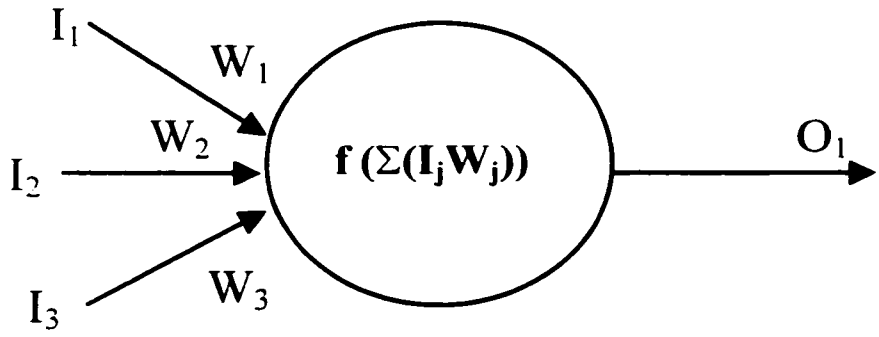


Figure 4.5 A model of artificial neuron

4.2.2 Self Organizing Networks

Kohonen describes a speaker adaptive system using an unsupervised learning algorithm [14]. Kohonen's self-organizing map (SOM) networks are designed to learn relations in an unsupervised manner. After training, the network is able to group similar inputs together in the output layer.

As the SOM is unsupervised, its performance may be improved using a supervised training method called learning vector quantization (LVQ) [15]. The main difference is that LVQ is concerned with searching for good category boundaries, while the SOM focuses on finding the reference vectors that are centroids of the input vectors. There are three types of LVQ: LVQ1, LVQ2 and LVQ3 [14]. In LVQ, the input data must be labeled and the outputs are divided into different classes. The learning rule is based on moving the winning weight vector toward the corresponding input vector. Eventually, the weight vector will become close representations of the input vectors after training. These weights vectors forms a trained weight matrix called codebook.

The architecture of a LVQ neural network is shown in Figure 4.6. Since the motivation of LVQ algorithm is to find the output unit that is the closest to the input vector, the vectors in codebook are adjusted according to the input vector. If input vector **x** and a reference vector belong to the same class, then the weights are moved toward the new

input vector: if $\mathbf{x}$ and $\mathbf{w}_j$ belong to different classes, then we move the weights away from the input vector. The algorithm is summarized as:

(1) Initialize the codebook vectors and learning rate a(0)

(2) For each training input vector x, find the winner $\mathbf{w}_j$ so that $||\mathbf{x}-\mathbf{w}_j||$ is minimum

(3) Update $\mathbf{w}_j$ as follows:

if $\mathbf{x}$ and $\mathbf{w}_j$ belong to same class, then

$$\mathbf{w}_j(\text{new}) = \mathbf{w}_j(\text{old}) + a\,[\mathbf{x}-\mathbf{w}_j(\text{old})];$$

if $\mathbf{x}$ and $\mathbf{w}_j$ belong to different classes, then

$$\mathbf{w}_j(\text{new}) = \mathbf{w}_j(\text{old}) - a\,[\mathbf{x}-w_j(\text{old})]$$

(4) reduce learning rate

(5) if stopping condition is not satisfied, then repeat step 2 -- 4, otherwise stop.

In step 1, the codebook vectors could be initialized by either taking the first m training vectors or the vectors with random values.

LVQ2 and LVQ3 are two improved algorithms based on the LVQ1. In LVQ1, only the winning reference vector is updated during training. The moving direction is determined by whether the winning vector belongs to the same class as the input vector. In the improved LVQ algorithms, two vectors (the winner and the runner-up) will be updated if several conditions are satisfied.

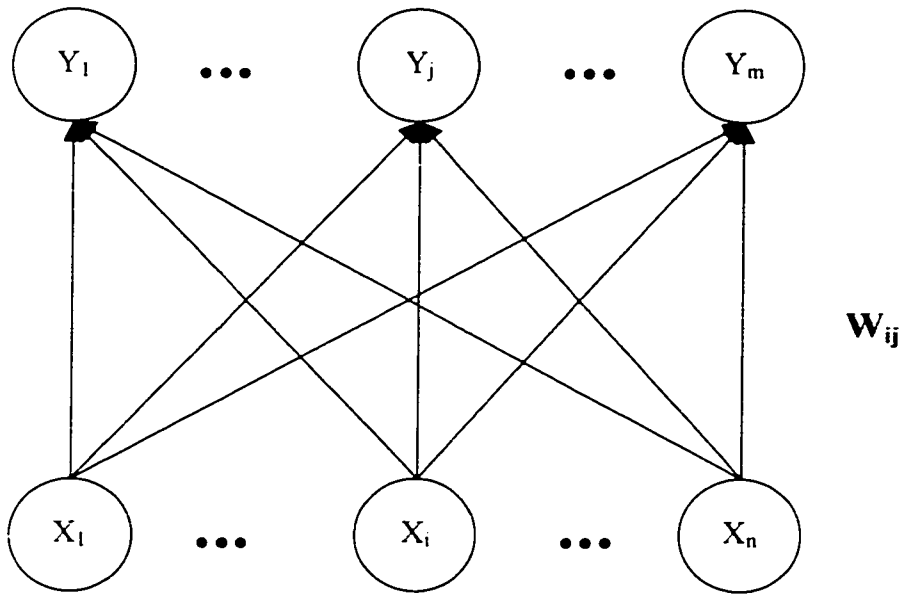


Figure 4.6 Learning vector quantization neural network

4.2.3 Fuzzy Neural System

The theories of fuzzy sets and neural networks are two complementary ways of modeling the human brain. Neural networks model the physical structure of the human neural network, while fuzzy logic simulates the way of human thinking. Therefore, the combination of fuzzy sets and neural networks, which is called fuzzy neural networks, are becoming very promising for exploring the human brain.

Considering the role and interaction of fuzzy logic and neural networks, researchers are studying various issues on combining them for various applications, such as fuzzy reasoning and pattern recognition. Currently, fuzzy systems are beginning to recognize the use of neural network in various aspects of reasoning. A successful example of the combination is fuzzy learning vector quantization (FLVQ) [21].

FLVQ has similar structure with LVQ. It extends the LVQ algorithm with fuzzy concepts. In LVQ, the principle of updating is basically "winner takes all". In other words, the winner obtains a complete membership 1, while all the others get 0. Even in LVQ2 and LVQ3, the membership is only given to the winner and the runner-up. Based on this, learning is only applied to update one or two reference vectors. In contrast, all the reference vectors are updated in FLVQ. For a specific training vector, FLVQ assigns various membership degrees to all the reference vectors, which provides the detailed learning information.

Assuming c is the number of classes (i.e., the dimension of the second layer), the FLVQ algorithm is described as follows [21]:

(1) generate an initial set of reference vectors $\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, , \ldots , \mathbf{w}_c\}$, select m_i and m_f as the initial and final values for the fuzziness parameter m; set the iteration number p = 0 and N as the maximum number of iterations;

(2) set $m = m_i + p\ [(m_f - m_i) / N]$, calculate the membership degrees between *ith* training vector and *jth* weight vector:

$$u_{ij} = \left[\sum_{l=1}^{c} \left(\frac{\| x_j - w_i \|^2}{\| x_j - w_l \|^2} \right)^{1/(m-1)} \right]^{-m} \qquad 1 \le i \le c; \quad 1 \le j \le n \tag{4.1}$$

(3) Update reference vectors:

$$w_i(\text{new}) = w_i(\text{old}) + a_i \sum_{j=1}^{n} u_{ij}[x_j - w_i(\text{old})], \quad 1 \le i \le c \tag{4.2}$$

where learning rate a_i is

$$a_i = \frac{1}{\sum_{j=1}^{n} u_{ij}} \tag{4.3}$$

(4) if stopping condition is not satisfied, then repeat step 2 -- 3, otherwise stop.

4.3 Fuzzy C-Means Clustering

Fuzzy C-Means (FCM) is a data clustering technique where each data belongs to a cluster with a degree specified a membership degree. The technique was originally introduced by Jim Bezdek [21] in 1981 as an improvement of earlier clustering methods [21]. In the following sections, the algorithm and application of fuzzy c-means clustering for speech recognition will be described.

4.3.1 Algorithm of FCM

Assuming there are n vectors $\mathbf{x}_i$ with i = 1, 2, ... , n, then fuzzy C-means clustering will partition the feature vectors $\mathbf{x}_i$ into c fuzzy groups, and find a cluster center for each group to minimize an objective function of dissimilarity. All cluster centers are represented by a prototype matrix V = (v_1, v_2, ... , v_c). To accommodate the introduction of fuzzy clustering, the membership matrix U = $\{u_{ij}\}$ is generated with the values of each element set to be between 0 and 1. Thus, the summation of all membership degrees for each cluster center was guaranteed to be equal to unity because of the normalization property:

$$\sum_{i=1}^{c} u_{ij} = 1, \qquad j = 1, 2, \ldots, n \tag{4.4}$$

The objective function for FCM is defined as

$$O(U, v_1, v_2, \ldots, v_c) = \sum_{i=1}^{c} o_i = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2}, \tag{4.5}$$

where u_{ij} is the element of membership matrix U which should have value between 0 and 1, v_i stands for the cluster center (or prototype) of the fuzzy group i, $d_{ij} = \| v_i - \boldsymbol{x}_j \|$ is the Euclidean distance between ith cluster center and jth input vector, and m is a weighting parameter which indicates the degree of fuzziness. The parameter m is usually set as a real value greater than 1.

The necessary conditions to minimize the objective function $\boldsymbol{O}$ in equation 4.5 can be found by forming a new objective function $\boldsymbol{O'}$ as:

$$\begin{aligned} & O'(U, v_1, v_2, \ldots, v_c, \lambda_1, \lambda_2, \ldots, \lambda_n) \\ & = O(U, v_1, v_2, \ldots, v_c) + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{c} u_{ij} - 1 \right) \\ & = \sum_{i=1}^{c} \sum_{j=1}^{n} u_{ij}^{m} d_{ij}^{2} + \sum_{j=1}^{n} \lambda_j \left(\sum_{i=1}^{c} u_{ij} - 1 \right), \end{aligned} \tag{4.6}$$

where λ_j (j = 1 to n) are the Lagrange multipliers for the n constraints. By differentiating $\boldsymbol{O'}$ with respect to each of its input arguments, the necessary conditions to minimize the objective function are:

$$v_j = \frac{\sum_{j=1}^{n} (u_{ij})^{m} x_j}{\sum_{j=1}^{n} (u_{ij})^{m}}, \qquad 1 \le i \le c \tag{4.7}$$

and

$$u_{ij} = \left[\sum_{l=1}^{c} \left(\frac{\| x_j - v_i \|^2}{\| x_j - v_l \|^2} \right)^{1/(m-1)} \right]^{-1} \qquad 1 \le i \le c; \quad 1 \le j \le n \tag{4.8}$$

Based on the above analysis, the FCM algorithm is simply an iterative procedure to meet the above two necessary conditions to minimize the objective function. Initially, the cluster centers are very inaccurately placed, and every data point has a membership grade for each cluster. By iteratively updating the cluster centers and the membership grades for each data point, the cluster centers can be moved to the right location in order to minimize the objective function that represents the distance from any given data point to a cluster weighted by its membership grade. After these batch procedures, the cluster center and membership matrix will eventually be determined. FCM algorithm can be summarized as follows [22]:

(1) Select c, n, and e as a tolerance value for the objective function; set fixed number N as the maximum epoch and iteration counter $q = 0$.

(2) Initialize the cluster center $V_0 = \{v_{1,0}, v_{2,0}, \ldots, v_{c,0}\}$ for the first iteration;

(3) Set $q = q + 1$, and update the membership degree, the cluster center and convergent variance as follows:

$$u_{ij,q} = \left[\sum_{l=1}^{c} \left(\frac{\| x_j - v_{i,q-1} \|^2}{\| x_j - v_{l,q-1} \|^2} \right)^{1/(m-1)} \right]^{-1} \qquad 1 \le i \le c; \quad 1 \le j \le n \tag{4.9}$$

$$v_{i,q} = \frac{\sum_{j=1}^{n} (u_{ij,q})^{m} x_j}{\sum_{j=1}^{n} (u_{ij,q})^{m}}, \quad 1 \le i \le c \tag{4. 10}$$

$$E_q = \sum_{i=1}^{c} \| v_{i,q} - v_{i,q-1} \|^2 \tag{4. 11}$$

(4) If $q < N$ and $E_q > e$, then go to step 3.

4.3.2 An Example

To illustrate how fuzzy c-means clustering works, let's have a simple example with the two-dimensional data that belong to two classes. Figure 4.7 (a) plots out all the 16 two-dimensional data.

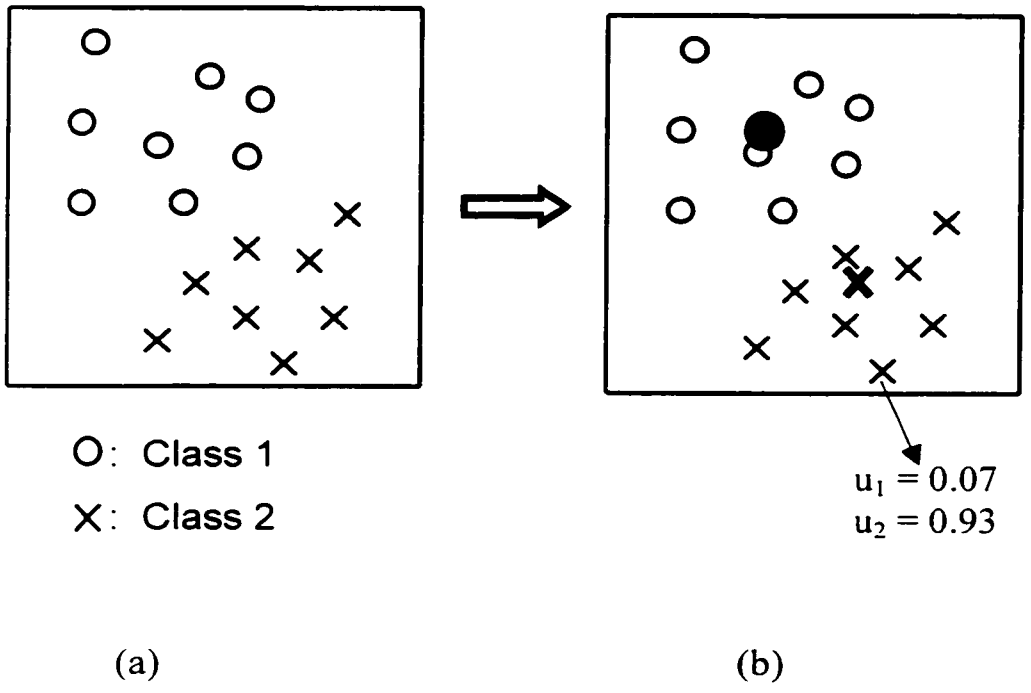


Figure 4.7 (a) Two-dimension data before clustering

(b) The cluster centers found by FCM

After applying fuzzy c-means clustering algorithm, two centers were located with the bigger symbol as shown in Figure 4.7 (b). Each data point has a membership grade for the two cluster centers. For instance, the bottom-right point has a member grade 0.07 for cluster 1, and 0.93 for cluster 2.

4.3.3 Summary

FCM can be applied to various clustering applications. In this thesis, FCM is used for clustering the speech features for a certain number of isolated words during the training process. For training each word, a number of samples from different speakers are chosen to form the template. As we know, there are many factors to cause the variability between different samples for even the same words. Therefore, FCM can be used to searching the cluster center for each word.

Chapter 5

Fuzzy Speech Recognizer

5.1 Issues on Implementing a Fuzzy Speech Recognizer

5.1.1 Time Normalization

When implementing a speech recognition system, a speech pattern is usually represented by a spectral sequence on a short-time basis. In most pattern recognition techniques, These spectral sequences will be compared in order to decide the matching score. However, if a word is spoken twice by the same speaker under the same environment, it is still very likely that the two samples will have different lengths. The main reason of this is that different renditions of the same utterance are seldom pronounced at exactly the same speed and manner across the whole utterance. To deal with the speaking rate fluctuation, it is strongly required to normalize the speech signal in order to make comparison and decision between patterns.

In the traditional algorithms, one of the waveforms is warped onto the time axis of the other one. Consider two speech patterns X and Y which are represented by $(x_1, x_2, \ldots x_{Tx})$ and $(y_1, y_2, \ldots y_{Ty})$, where x_i and y_i stand for the short-time feature vectors and T_x, T_y denote the duration of Pattern X and Y respectively. In real applications, the duration T_x

and T_y usually have different values. The dissimilarity between X and Y should be measured based on solving the problem of normalizing the two sequences into the same lengths.

In this thesis, the linear time normalization method is used for pattern recognition. The dissimilarity between pattern X and Y is defined as:

$$d(X,Y) = \sum_{i_x=1}^{T_x} d(x_{i_x}, y_{i_y}) \tag{5.1}$$

Where i_x and i_y are integer numbers which denote the time indices of X, Y; and $d(x_{ix}, y_{iy})$ is a function for dissimilarity measurement between two vectors. Also, i_x and i_y should satisfy the following constraints:

$$i_y = \frac{T_y}{T_x} i_x \tag{5.2}$$

By rounding the length of pattern Y to the same length as pattern X, the summation of distance for each vector in equation (5.1) is defined as the dissimilarity of X and Y. Depending on the direction of the time normalization, the summation can be taken from $i_y = 1$ to T_y as well. Figure 5.1 illustrates how linear time normalization works for the index conversion.

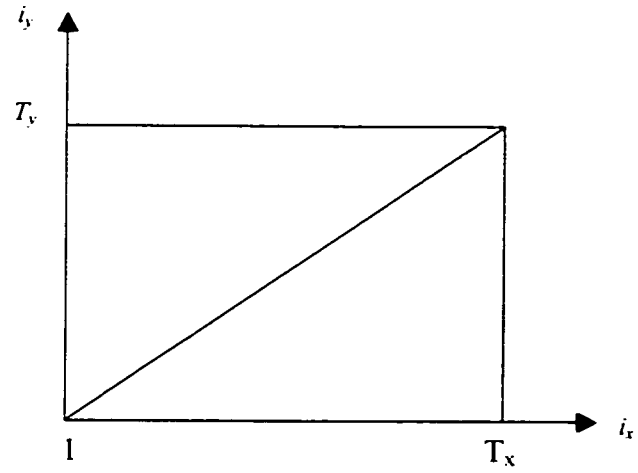


Figure 5.1 Linear time normalization for two sequences with different length

$$1, 2, \ldots T_y \rightarrow 1, 2, \ldots T_x$$

5.1.2 *Template Training*

The template-based method is used to implement the recognition system in this thesis. As shown in Figure 5.2. The feature vectors of an unknown word are fed into the recognition network as the input. By computing the dissimilarity between the input feature and each speech template, the network can eventually decide the identity of the unknown word with the decision algorithms.

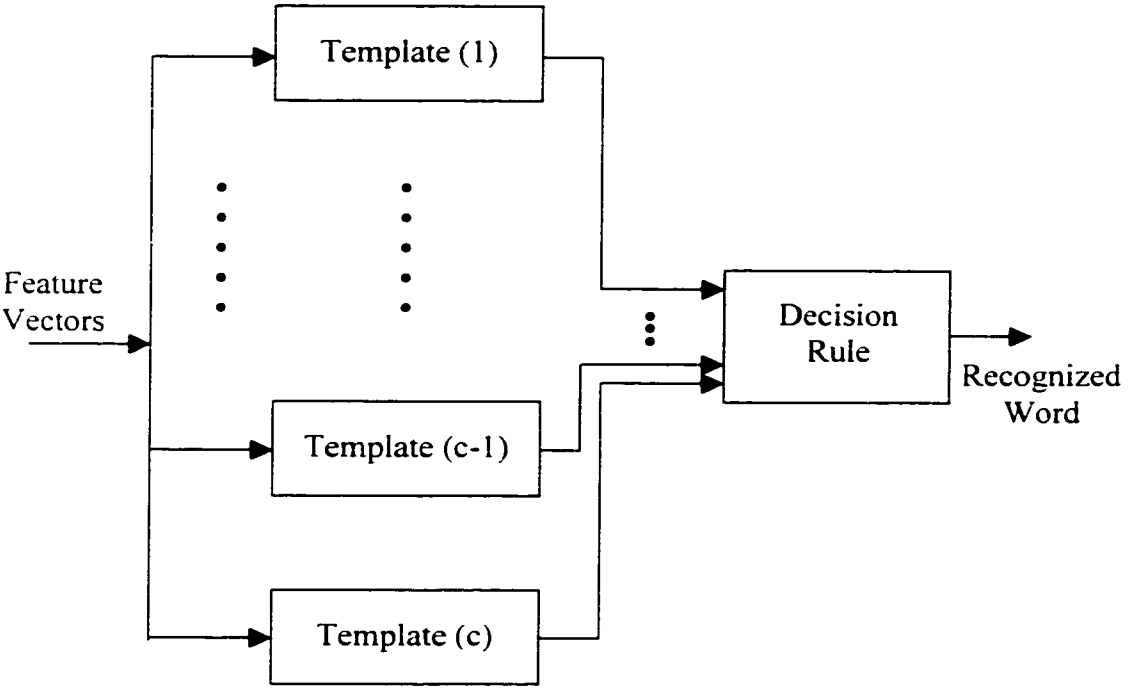


Figure 5.2 Template-based word recognition system

Before applying the pattern comparison technique according to Figure 5.2, firstly the templates should be trained and saved into a group of buffers which act like a memory storing the related "dictionary". Assuming there are totally c words in the recognition library, it means that c templates need to be trained, where each word template is represented by the time-frequency feature.

Because the decision results rely on the templates very much, it is very critical to obtain high quality templates that could represent the word features accurately. As described in Chapter 2, the difficulties of speech recognition are mainly caused by all kinds of invariance of speech signals. Therefore, the ideal templates should be able to model and include the time-frequency information of the speech signal with all the possible fluctuations during training such as:

- Speaker fluctuation
- Different speaking rate
- Different manner of utterance
- Environment noise

However, it is an extremely difficult task to take care of all the fluctuations in a real implementation. Based on the fact that the most important variations are the speaker fluctuation and speaking rate fluctuation, the clustering method will concentrate on dealing with these two problems. Therefore, the training sets should contain the speech signal taken from several speakers with different speaking rates.

The classical methods for template training include hard c-means clustering, self-organizing map, and LVQ etc.. In this thesis, the FCM algorithm is used for clustering the training samples and locating template centers because it offers the advantage of modeling the speech fluctuations efficiently.

5.1.3 Recognition Network

In the experiments, recognition is performed using the fuzzy neural techniques for pattern matching. The membership functions are trained and used as the network weight. Two networks are developed based on measuring the similarity and dissimilarity respectively. More details of these two methods are introduced in the following sections.

Network 1

The basic idea of the fuzzy networks is to use the membership function for classifying the word patterns that consist of the time-frequency feature. To illustrate the theory, let's start from a simple example based on the typical parameter of vowels - formant frequencies. The formants are defined as the resonant frequencies of the vocal tract, and it is known that the first three formant frequencies could decide the characteristics of a vowel. Therefore, the membership functions should have three peaks, with each peak correspond to one formant. To generalize the membership function, the peak values of

membership function are normalized by 1/3 (Figure 5.3). If all formants of an unknown pattern can match the peaks of a membership function exactly, then the membership degree should be one. On the other hand, if the unknown pattern doesn't match the membership function or has shift from the center, it should get low membership degree. The degree D is denoted by:

$$D = \int_0^{\infty} m(f) \times y(f) df \tag{5.3}$$

Where y(f) indicates the location of formants f_1, f_2, f_3 as:

$$y(f) = h(f - f_1) + h(f - f_2) + h(f - f_3) \tag{5.4}$$

$$h(f) = \begin{cases} 1 & for \quad f = 0 \\ 0 & for \quad f \neq 0 \end{cases} \tag{5.5}$$

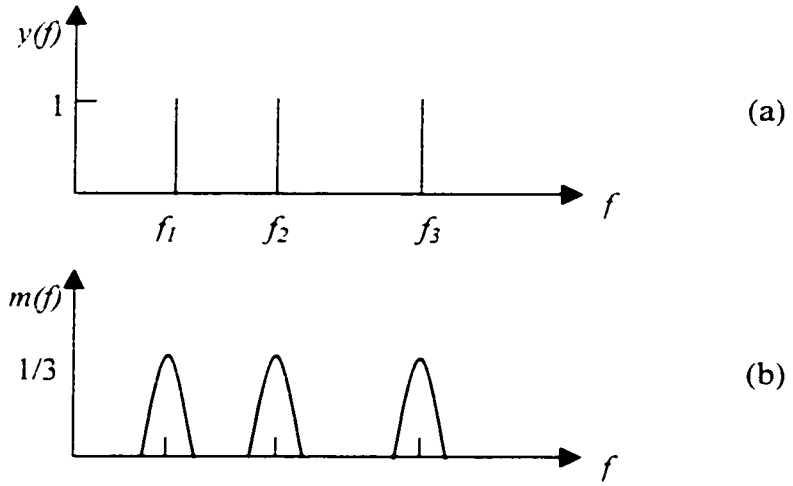


Figure 5.3 (a) Formant frequencies of a vowel

(b) Membership function of a vowel

Because line spectrum frequencies can provide the formant information, LSFs are used to form the feature vectors and membership functions in the recognition network. Assuming the order of LPC model is 10, then there are 10 line spectrum frequencies F = $\{f_1, f_2, \ldots, f_{10}\}$. The membership function can be constructed with rectangular or Gaussian-shaped function as shown in Figure 5.4. The Gaussian function in term of f_1 and f_2 is given by:

$$G(f) = e^{-\frac{(f-\alpha)^2}{2\delta^2}} \qquad (5.6)$$

where $\alpha = (f_1+f_2) / 2$, $\delta = (f_2 - f_1) / 2$.

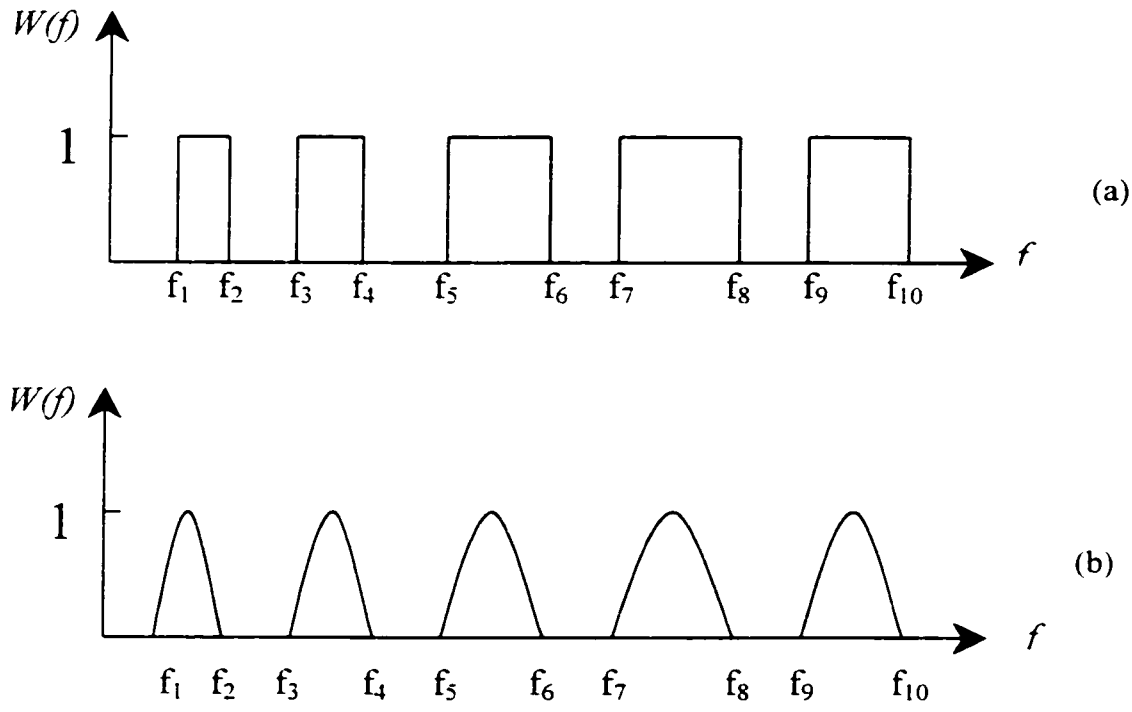


Figure 5.4 (a) Rectangular shape membership function

(b) Gaussian-shaped membership function

The input vector **x(f)** of a speech frame is also constructed by LSFs in rectangular shape. In the recognition network as shown in Figure 5.5, similarities between the unknown feature **x** and the template patterns are firstly calculated, then the unknown pattern is classified into the category which gets the largest similarity score.

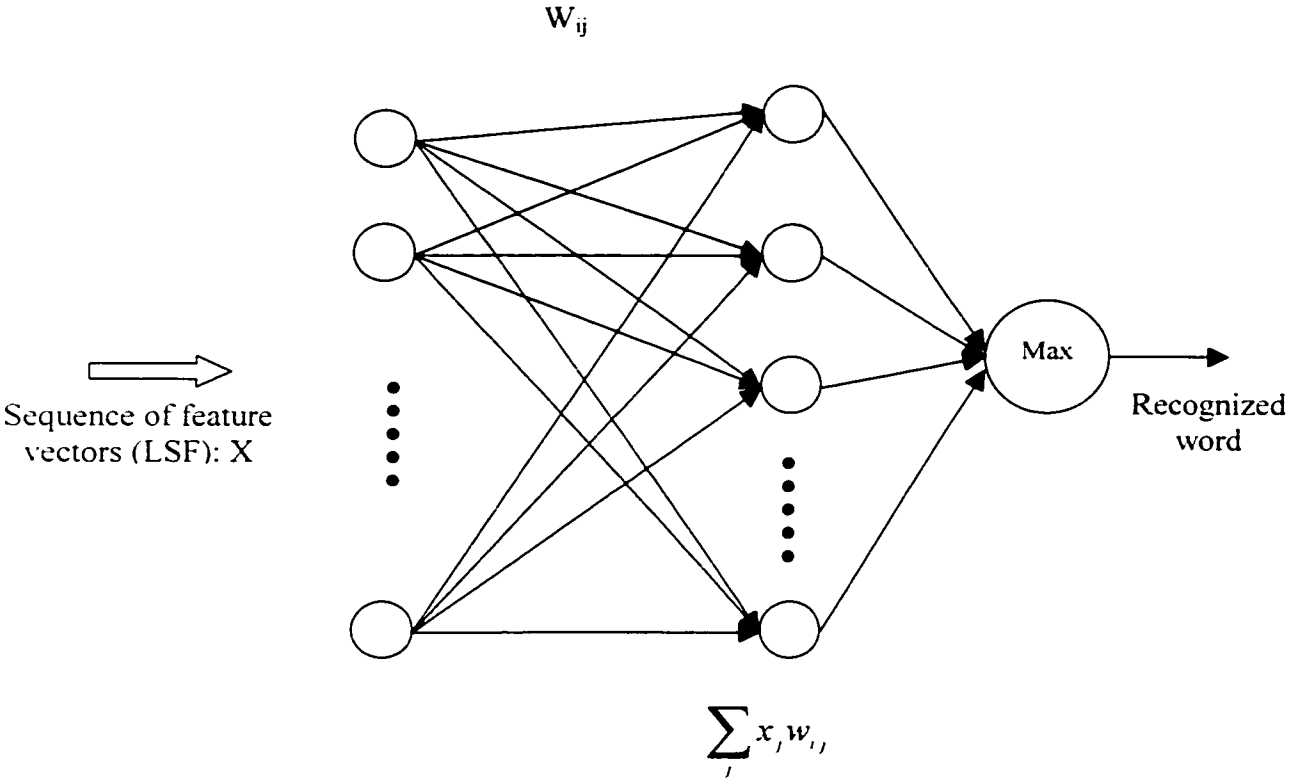


Figure 5.5 Fuzzy neural network for isolated word recognizer based on similarity measurement (Network 1)

Network 2

Network 2 (Figure 5.6) has a similar structure to that of network 1, but they are based on different decision rules. More specifically, network 1 measures the similarity between the unknown and the template patterns, then recognizes the word with the maximum similarity; while network 2 measures the dissimilarity or distance and takes the minimum as the winner.

Because network 1 is based on matching the information of formant frequencies between the unknown and the templates, only the line spectrum frequencies are appropriate to be used as the time-varying feature for it. In network 2, more coefficients could be adopted for speech characteristics, such as cepstrum, log area ratio, reflection coefficients, etc.

When an unknown feature matrix **X** is applied to the network, the recognition process is summarized as follows:

(1) Normalize the length of the unknown pattern into the same length as each template weight;

(2) Calculate the value of dissimilarity between the unknown and all templates frame by frame;

(3) Recognize the unknown word as the pattern which gets the smallest dissimilarity.

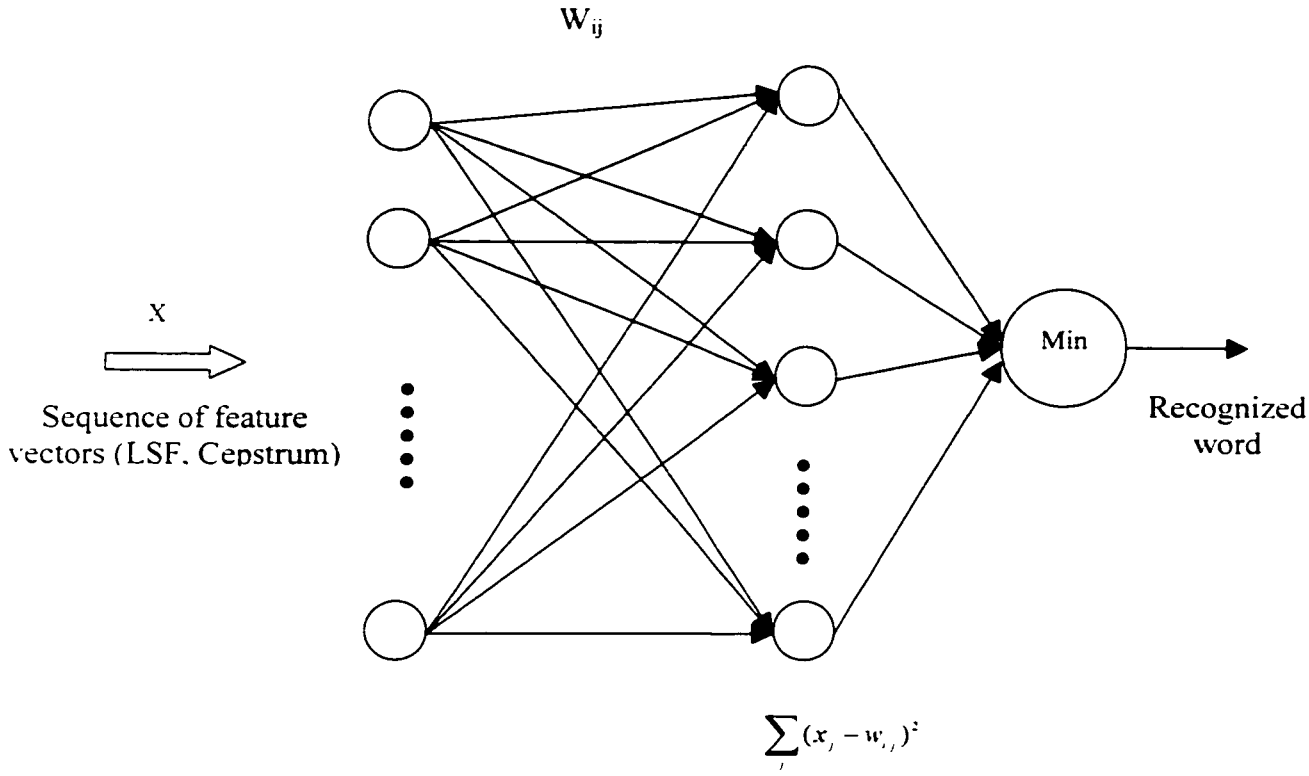


Figure 5.6 Fuzzy neural network for isolated word recognizer based on dissimilarity measurement

5.2 Speech Database

The speech database used for the recognition experiment consists of 10 isolated English words. All the ten words are recorded with 8kHz sampling rate, 16-bit quantization precision under laboratory environment. Each word is recorded ten times by ten speakers (6 male and 4 female). Consequently, the speech database has a total number of 1000 utterances, in which there are 100 utterances for each speaker.

5.3 Simulations and Results

In this thesis, the line spectrum frequencies and LPC cepstral coefficients are used as the speech feature sets. Both speaker dependent and speaker independent recognition are tested in the experiment.

Before processing, endpoint detection is performed for each utterance. Then the speech signals are pre-emphasized and blocked into small frames with 10ms overlapping between adjacent frames. The pre-emphasis factor is set to 0.95. For each frame, the Hamming window is applied with 30ms window length; and then speech feature sets are extracted based on the algorithm of LSF and LPC cepstrum.

In speaker-dependent recognition, the training data consist of 600 utterances from 6 speakers, and the remaining 400 utterances from these 6 speakers are used for testing. Table 5.1 shows the speaker-dependent recognition rate with the techniques described above.

Table 5.1 Recognition rate for speaker-dependent recognition

	Network 1		Network 2 (LSF)		Network 2 (Cepstrum)		Network 2 (Weighted Cepstrum)	
	Crisp	FCM	Crisp	FCM	Crisp	FCM	Crisp	FCM
Wayne	21/30	23/30	28/30	28/30	26/30	28/30	26/30	28/30
John	18/30	21/30	28/30	29/30	30/30	29/30	27/30	28/30
Walter	27/30	26/30	29/30	29/30	29/30	29/30	30/30	29/30
Hari	23/30	25/30	29/30	29/30	29/30	29/30	29/30	29/30
Tracy	28/30	28/30	30/30	30/30	30/30	30/30	30/30	30/30
Halima	29/30	29/30	30/30	28/30	30/30	30/30	30/30	30/30
Yes	20/30	21/30	25/30	28/30	25/30	26/30	27/30	28/30
No	23/30	24/30	29/30	29/30	23/30	24/30	26/30	27/30
Call	21/30	21/30	30/30	30/30	27/30	29/30	26/30	28/30
Hangup	28/30	29/30	30/30	30/30	30/30	30/30	30/30	30/30
Overall	79.3%	82.3%	96%	97%	93%	94.7%	93.6%	95.7%

For comparison, Figure 5.7 gives the overall recognition rate with FCM and crisp-mean for all the methods. It is shown that FCM performs better than when taking the crisp mean value as templates.

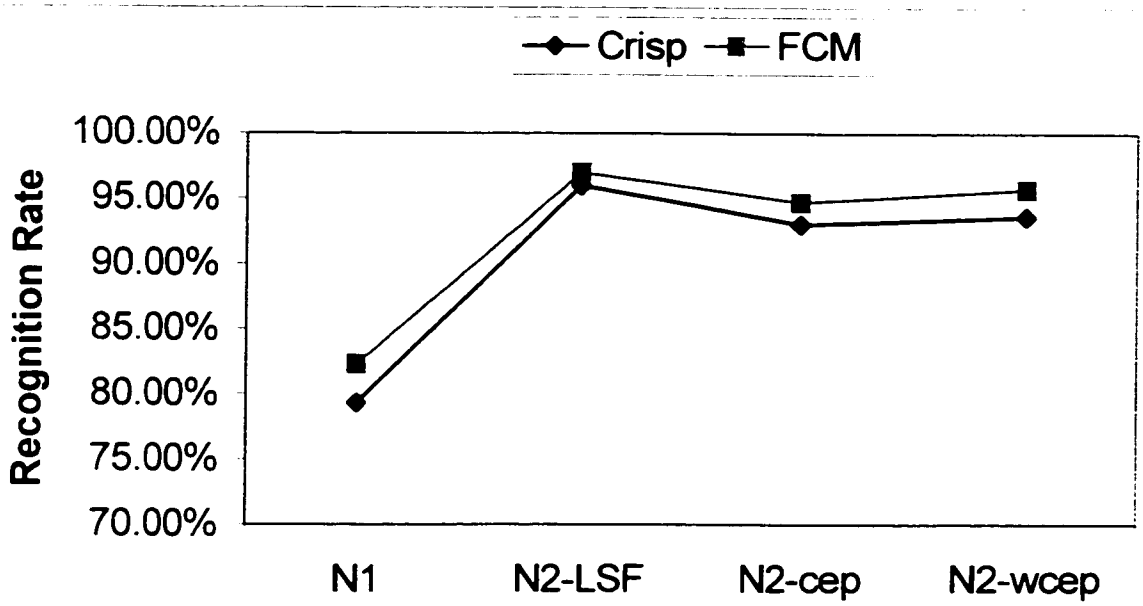


Figure 5.7 Comparison of the speaker dependent recognition rate with FCM and crisp means

It is shown in Figure 5.8 that network 2 yields better recognition rates than network 1 because the dissimilarity is utilized for decision making, which should be more accurate for distinguishing confusing words than when similarity measurement is used.

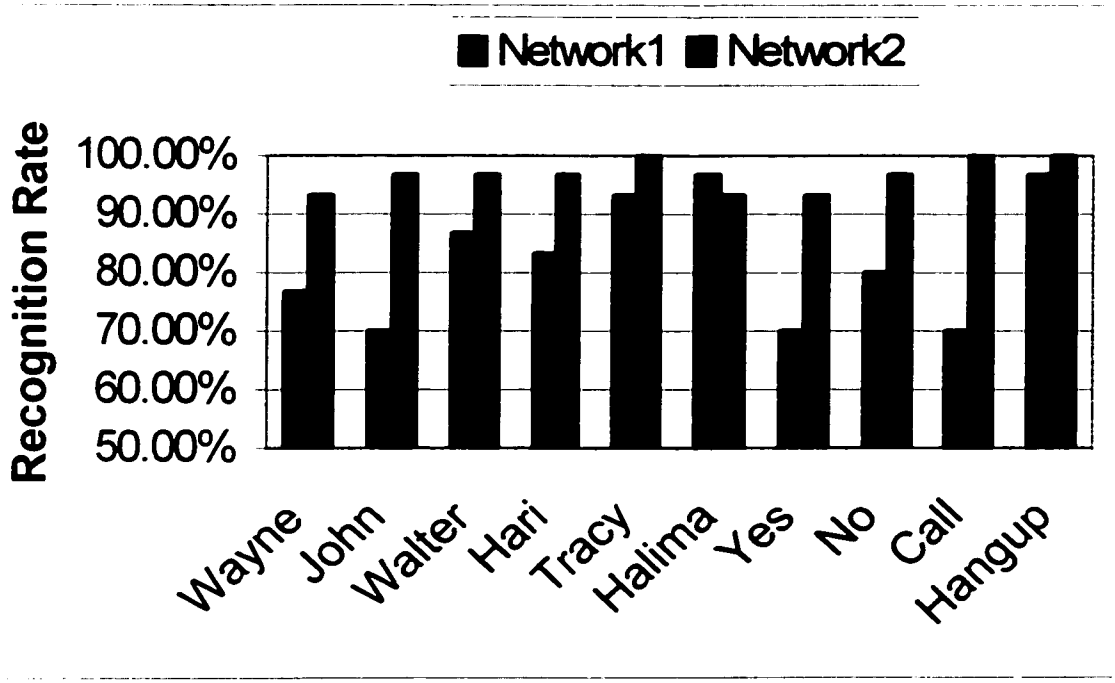


Figure 5.8 Speaker dependent recognition rate using LSF with network 1 and network 2

Speaker-independent recognition uses 600 utterances from 6 speakers (3 female and 3 male). The remaining 400 utterances from other 4 speakers are used as test data. Table 5.2 shows the recognition accuracy for all the words.

Table 5.2 Recognition rate for speaker-independent recognition

	Network 1		Network 2 (LSF)		Network 2 (Cepstrum)		Network 2 (Weighted Cepstrum)	
	Crisp	FCM	Crisp	FCM	Crisp	FCM	Crisp	FCM
Wayne	33/40	34/40	38/40	38/40	37/40	39/40	35/40	38/40
John	19/40	25/40	30/40	31/40	28/40	33/40	29/40	31/40
Walter	35/40	34/40	38/40	39/40	38/40	40/40	36/40	39/41
Hari	31/40	30/40	37/40	37/40	36/40	36/40	38/40	37/40
Tracy	34/40	35/40	32/40	36/40	31/40	36/40	38/40	40/40
Halima	35/40	36/40	34/40	36/40	30/40	35/40	35/40	36/40
Yes	16/40	18/40	34/40	34/40	31/40	32/40	28/40	35/40
No	26/40	27/40	35/40	35/40	36/40	34/40	32/40	35/40
Call	27/40	28/40	31/40	31/40	30/40	34/40	31/40	32/40
Halima	38/40	36/40	39/40	38/40	40/40	38/40	38/40	39/40
Overall	73.5%	75.8%	87%	88.8%	84%	89.3%	85.3%	90.5%

In speaker-independent recognition, it is also proved that FCM yield better result than crisp mean for template training (Figure 5.9). Figure 5.10 gives the comparison of network 1 and network 2 using LSF as speech features.

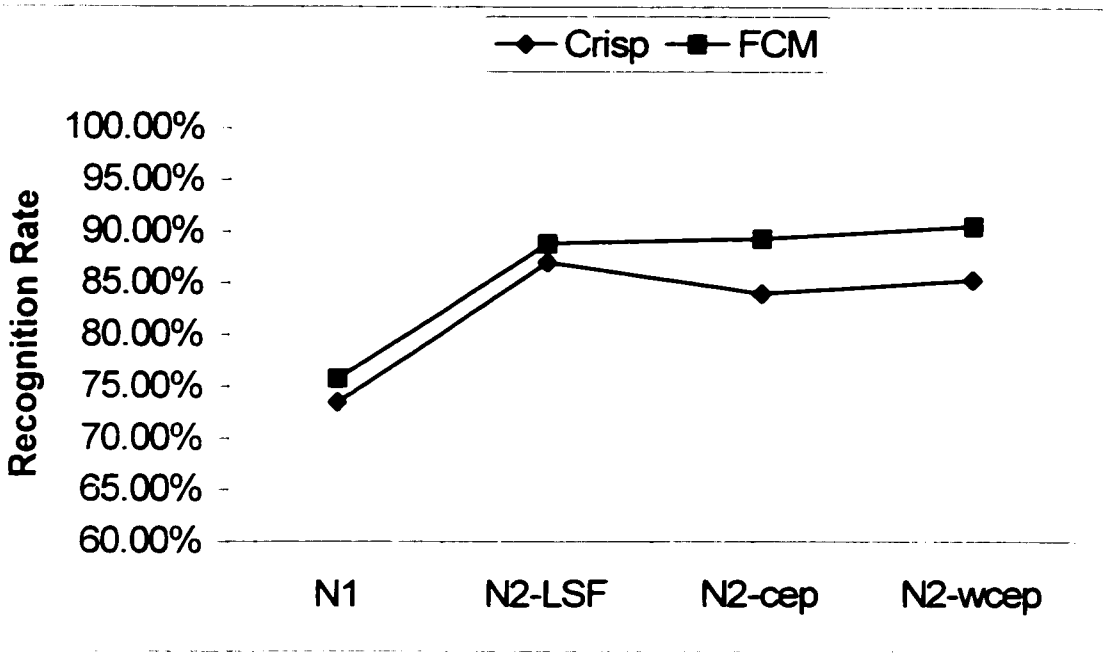


Figure 5.9 Speaker-independent recognition rate with FCM and crisp means

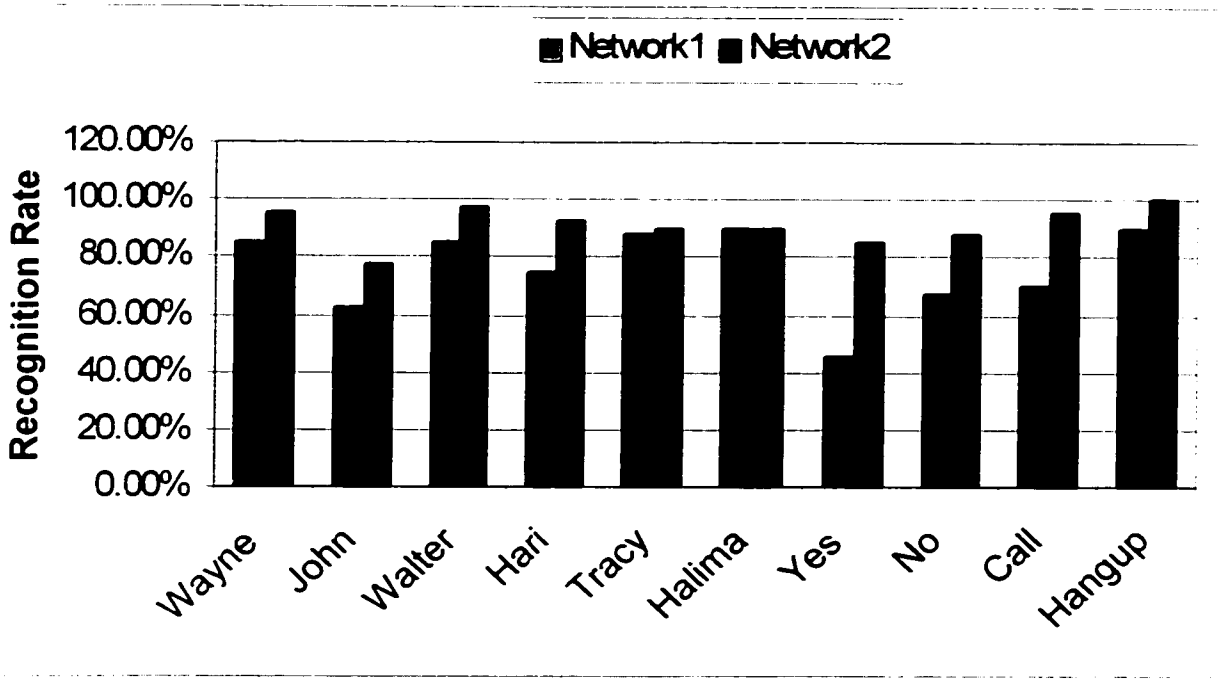


Figure 5.10 Speaker-independent recognition rate using LSF with network 1 and network 2

Chapter 6

Conclusions and Suggestion for Future Work

6.1 Conclusions

This thesis explored the issues involved in designing an isolated word recognition system, especially the application of fuzzy neural algorithms for speech pattern recognition. The LPC speech analysis method is described, and different representation parameters are compared. It is shown that the cepstral coefficients and line spectrum frequencies play important roles as speech features in recent research and applications of speech processing.

Naturally, fuzzy logic is similar to the way of human thinking. Fuzzy sets are successfully applied for speech recognition due to their ability to deal with uncertainty. However, there's always a balance between "fuzzy" and "too fuzzy". The idea of "fuzzy" is good for modeling the uncertainty and variance of speech signals. But if it's "too fuzzy", it is highly probable that it will cause a lot of confusion between the patterns which are similar to each other but actually different. For instance, the word "bad" and "bed" are very similar to each other, and fuzzy logic may not be distinguishable enough for this case. Therefore, the neural networks are introduced to incorporate with fuzzy logic to overcome this problem. As we know, neural networks simulates the "hardware"

of the human brain (human nerve) and have been known as a technique with great advantages of fault tolerance and robustness.

In this thesis, two fuzzy networks have been proposed and applied for isolated word recognition. The membership functions are constructed by means of superposition of speech features for each enrolled word and the templates are learned based on the membership functions. The template includes the fluctuation information of frequency and time. By using these templates, the recognition system is able to recognize spoken words independent of the speaker.

The analysis and results of the recognition technique reveals that the use of fuzzy logic and neural networks can consistently improve the performance of the system. From the results, FCM has been shown to be a better template-training algorithm than hard clustering.

6.2 Suggestions for Future Work

The results in the thesis have proved the potential of fuzzy theory and neural networks for speech recognition. Based on the proposed methods, It is still possible to improve the system and get higher recognition rate.

An important issue in the fuzzy neural network area is to find efficient combinations of ANNs inspired by the structure of the human cortex because it forms the most intelligent speech recognizer so far. Also, it is certainly a promising direction to simulate the natural model of speech perception and production, for both the feature extraction and pattern recognition part.

Since some other techniques have already been successfully used for speech recognition, more efficient and integrated systems could be constructed by combining fuzzy neural techniques with other formalisms, such as HMMs and DTW.

References

[1] Jean-Claude Junqua, Jean-Paul Haton, *Robustness in automatic speech recognition fundamentals and applications*, Kluwer Academic Publishers, 1996.

[2] Lawrence Rabinar, Bing-hwang Juang, *Fundamentals of speech recognition*, Prentice Hall, Englewood Cliffs, 1993.

[3] Joseph P. Campbell, JR., "Speaker recognition: A tutorial", *Proceedings of IEEE*, Vol. 85, No. 9, September 1997.

[4] Lawrence Rabinar, Ronald W. Schafer, *Digital processing of speech signals*, Prentice Hall, Inc., Englewood Cliffs, NJ, 1978.

[5] C. H. Chen, *Fuzzy logic and neural network handbook*, McGraw-Hill Inc., 1996.

[6] F. Itakura, "Line spectrum representation of linear predictive coefficients of speech signals", *J. Acoust. Soc. Amer.* Vol. 57, pp. 535(a), 1975.

[7] K. K. Paliwal, "A study of line spectrum pair frequencies for speech recognition", *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing 1988*, Vol. 1, pp. 485 – 488.

[8] Samir Saoudi, Jean Marc Boucher, "A new efficient algorithm to compute the LSP parameters for speech coding", *Signal Processing*, pp 201-212, 1992.

[9] Seung Ho Choi, Hong Kook Kim, Hwang Soo Lee and R. M. Gray, "Speech recognition method using quantised LSP parameters in CELP-type coders", *Electronics Letters*, 22 October, 1997.

[10] K. K. Palwal, "A study of line spectrum pair frequencies for vowel recognition", *Speech Communications* 1999, pp. 27-33.

[11] Chi-Shi Liu, Chao-Shih Huang, Min-Tau Lin and Hsiao-Chuan Wang, "Automatic speaker recognition based upon various distances of LSP frequencies", *IEEE International Carnahan Conference on Security Technology Oct., 1991*, pp. 104-109.

[12] Frank K. Soong, Biing-Hwang Juang, "Optimal Quantization of LSP parameters", *IEEE Transactions on Speech ans Audio Processing*, Vol. 1, No.1, January 1993.

[13] N. Naja, J.M. Boucher and S. Saoudi, "Fast LSP vector quantization algorithms comparison", *MELECON Proceedings of the 7th Mediterranean Electrotechnical Conference - MELECON.*, Part 3, Apr. 1994, pp. 1127 – 1130.

[14] Teuvo Kohonen, "The self-organizing Map", *Proceedings of the IEEE*, Vol. 78, No. 9, September 1990, pp. 1464 – 1477.

[15] Eeik McDermontt and Shigeru Katagiri, "LVQ-based shift-tolerant phoneme recognition", *IEEE Transactions on Signal Processing*, Vol. 39, No. 6, June 1991, pp. 1398 – 1410.

[16] Ravi P. Ramachandran, Mihailo S. Zilovic, and Richard J, Mammone, "A comparative study of robust linear predictive analysis methods with applications to speaker identification", *IEEE Transactions on Speech and Audio Processing*, Vol. 3, No. 2, March 1995, pp. 117 – 125.

[17] Akio Amano et al., "On the use of neural networks and fuzzy logic in speech recognition", *IJCNN Int. Jt. Conference on Neural Networks*, Jun 18-22, 1989, pp. 301-305.

[18] Christopher Hale, CamQuynh Nguyen, "Voice command recognition using fuzzy logic", *Wescon Conference Record Proceedings of the 1995 Wescon Conference*, Nov 7-9 1995, San Francisco, CA, USA, pp. 608-613.

[19] Lynn Yaling Cai, Hon Keung Kwan, "Fuzzy classifications using fuzzy inference networks", *IEEE Transactions on Systems, Man, and Cybernetics -- Part B: Cybernetics*, Vol. 28, No. 3, June 1998, pp. 334-347.

[20] Hon Keung Kwan, Yaling Cai, Bin Zhang, "Membership function learning in fuzzy classification", *Int. J. Electronics*, 1993, Vol. 74, No. 6, pp. 845-850.

[21] Nicolaos B. Karayiannis, Jame C. Bezdek, "An integrated approach to fuzzy learning vector quantization and fuzzy c-means clustering", *IEEE Transactions on Fuzzy Systems*, Vol. 5, No. 4, November 1997, pp. 622-628.

[22] Jyh-Shing Roger Jang and Jiuann-Jyn Chen, "Neuro-fuzzy and soft computing for speaker recognition", *IEEE International Conference on Fuzzy Systems Proceedings of the 1997 6th IEEE International Conference on Fuzzy Systems FUZZ-IEEE'97. Part 2 (of 3) July 1997 v 2 Barcelona, Spain*, pp. 663 – 668.

[23] Jun-ichiroh Fujimoto, Tomofumi Nakatani and Masahide Yoneyyama, "Speaker-independent word recognition using fuzzy pattern matching", *Fuzzy Sets and Systems 32*, 1989, pp. 181-191.

[24] Liusheng Liu, Zhijian Li and Bingxue Shi, "Speech recognition based in fuzzy vector quantazation and fuzzy logic", *IEEE International Conference on Neural Networks v5 1995 Perth, Aust, IEEE Piscataway NJ USA*, pp. 2858-2862.

[25] Liusheng Liu, Zhijian Li and Bingxue Shi, "Segment matrix vector quantization and fuzzy logic for isolated-word speech recognition", *Proceedings of The International Symposium on Multiple-Valued Logic, 1995*, pp. 152 – 156.

[26] James W. Pitton, Kuansan Wang, and Bing-Hwang Juang, "Time-frequency analysis and auditory modeling for automatic recognition of speech", *Proceedings of IEEE*, Vol. 84, No. 9, September 1996, pp. 1199 – 1215.

[27] K. Davis, R. Biddulph, and S. Balashek, "Automatic recognition of spoken digits", *J. Acoustic. Soc. Am.*, 1952, 24: pp. 3-50.

[28] J. Suzuki and K. Nakata, "Recognition of Japanese vowels - Preliminary to the recognition of speech", *J. Radio Res. Lab.*, 1961, pp. 193-212.

[29] P. Denes, "The design and operation of the mechanical speech recognizer", *Journal of the British Institute of Radio Engineers*, 1959, pp. 211-229.

[30] T. Vintsyuk, "Speech discrimination by dynamic programing", *Kibernetika, Cybernatics*, pp. 81-88.

[31] P. Ladefoged, "The phonetic basis for computer speech processing", *Computer Speech Processing*,1985, pp. 3-27.

[32] L. A. Zadeh, "Fuzzy sets", *Inform. Control*, 1965, pp. 338-352.

Vita Auctoris

Name: Hui PING

Place of Birth: Jiangsu, China

Year of Birth: 1973

Education:

B. Eng.
Department of Electronic Engineering
Nanjing University of Aeronautics and Astronautics
Nanjing, China
1990 - 1994

M. A. Sc
Electrical and Computer Engineering
University of Windsor
Windsor, Ontario, Canada
1997 - 1999